

DEPARTAMENTO DE INGENIERÍA DE SISTEMAS Y AUTOMÁTICA

**UNIVERSIDAD CARLOS III DE MADRID**

ESCUELA POLITÉCNICA SUPERIOR

PROYECTO FIN DE CARRERA



**SISTEMA COLABORATIVO PARA LA  
DIVULGACIÓN DE INCIDENCIAS CIUDADANAS  
BASADO EN ANDROID**

Tutor: Alberto Brunete González

Director: Luis Pedraza Gómara

Autor: Julio Alberto Molinera Ariza

Titulación: Ingeniería Técnica Informática de Gestión

LEGANÉS, MADRID

SEPTIEMBRE 2012



Título: Sistema colaborativo para la divulgación de incidencias ciudadanas basado en Android.

Autor: Julio Alberto Molinera Ariza

Titulación: Ingeniería Técnica Informática de Gestión

Tutor: Alberto Brunete González

Director: Luis Pedraza Gómara

## **EL TRIBUNAL**

Presidente:

Secretario:

Vocal:

Realizado el acto de defensa y lectura del proyecto Fin de Carrera el de 2012 en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, acuerda otorgarle la CALIFICACIÓN de:

Fdo. Presidente

Fdo. Secretario

Fdo. Vocal





“No basta saber, se debe también aplicar.  
No es suficiente querer, se debe también hacer.”  
(Johann Wolfgang Goethe)



## **AGRADECIMIENTOS**

En primer lugar quiero agradecer a mi familia el apoyo que me ha dado y sin el cual mi formación universitaria no habría sido posible.

Quiero agradecer en segundo lugar el apoyo mostrado por Lourdes quien ha resultado ser un modelo de constancia y sacrificio, en el cual me fijaba en momentos de baja moral.

En tercer lugar quiero agradecer a todo el mundo (amigos, familiares, conocidos) que ha calificado de “Sagrada Familia” o “Obra del Escorial” a mi proyecto, porque aunque por un lado me sentaba mal que criticarán mi tardanza en su realización, me daba ánimos a continuar con él.

Agradecer también a todo el que ha compartido conmigo este camino universitario y que me ha puesto piedras o me las ha quitado de él. Un camino sin piedras hubiera sido muy poco interesante.

Por último agradecer a mi Director de proyecto Luis Pedraza, la posibilidad de poder empezar este proyecto y a mi tutor Alberto Brunete la oportunidad de finalizarlo. Sin su colaboración nada de esto hubiera sido posible.



## RESUMEN

El objetivo de este Proyecto Fin de Carrera es crear un sistema integral capaz de gestionar las incidencias de los ciudadanos de una determinada zona geográfica. Para ello, se ha desarrollado un sistema de colaboración ciudadana mediante el cual una comunidad de usuarios puede presentar su disconformidad ante distintas situaciones que se producen en los lugares que les rodean, ya sea su barrio, pueblo o ciudad.

Este sistema permite tanto la denuncia de las distintas quejas ciudadanas, así como la consulta a partir de los distintos campos que la forman o a través de su situación geográfica. Todo ello se ha desarrollado con una interfaz totalmente orientada al usuario, intuitiva y de fácil manejo, buscando la utilización de la aplicación por un gran número de personas para que ésta pueda tener una mayor repercusión social.

El sistema que se ha desarrollado se compone de un servidor, una aplicación móvil para terminales Android e interfaces con aplicaciones de terceros (Google Maps, Picasa) necesarias para facilitar el manejo de la aplicación.

## ABSTRACT

The aim of this Master Thesis Project is to create an integral system, which can manage the incidences of the citizens of a particular geographic area. To do this, it has been developed a citizen collaboration system where a community of users can publish their dissatisfaction with the different situations that occur in places that surround them (neighborhood, village, city, etc.).

This system allows both the reporting and the query of citizen complaints, organized by the different fields of the form or by geographical location. All this has been fully developed with a user-oriented interface, intuitive and easy to use, expecting the use of the application for a large number of people so that it can have a greater social impact.

The system that has been developed consists of a server, a mobile application for Android devices and interfaces with third party applications (Google Maps, Picasa) to facilitate the handling of the application.



## INDICE

CAPÍTULO 1.	DESCRIPCIÓN DEL PROYECTO.....	14
1.1	INTRODUCCIÓN.....	14
1.1.1	LA WEB 2.0 .....	14
1.1.2	FOROS, BLOGS, WIKIS, REDES SOCIALES .....	15
1.1.3	APLICACIONES DE COLABORACIÓN CIUDADANA.....	16
1.2	OBJETIVOS .....	16
1.3	MEDIOS UTILIZADOS .....	17
1.3.1	SERVIDOR .....	17
1.3.2	APLICACIÓN ANDROID .....	17
1.3.3	SERVICIOS WEB .....	17
1.4	ESTRUCTURA DE LA APLICACIÓN .....	18
1.5	ESTRUCTURA DEL DOCUMENTO .....	19
CAPÍTULO 2.	ESTADO DEL ARTE .....	20
2.1	INTRODUCCIÓN.....	20
2.2	ESTUDIO DE APLICACIONES SIMILARES .....	20
2.2.1	FOTODENUNCIA.ORG.....	20
2.2.2	FOTODENUNCIAS.NET .....	23
2.2.3	PONLE FRENO.....	26
2.2.4	CONCLUSIÓN .....	31
2.3	ESTUDIO DE LAS POSIBLES SOLUCIONES.....	32
2.3.1	PLATAFORMA DE DESARROLLO .....	32
2.3.2	BASE DE DATOS .....	35
2.3.3	CONEXIÓN DE LA APLICACIÓN CON LA BASE DE DATOS.....	37
2.3.4	SERVIDOR .....	37
2.3.5	SERVICIO WEB DE IMÁGENES.....	38
2.3.6	SERVIDOR DE MAPAS .....	40
CAPÍTULO 3.	ANÁLISIS Y DISEÑO DEL SISTEMA .....	41
3.1	CASOS DE USO .....	41
3.2	ESPECIFICACIÓN DE REQUISITOS .....	46
3.3	DIAGRAMA DE CLASES .....	55



3.3.1	DIAGRAMA DE CLASES DE LA BASE DE DATOS.....	55
3.3.2	DIAGRAMA DE CLASES DE LA APLICACIÓN ANDROID .....	56
3.4	DIAGRAMAS DE SECUENCIA.....	57
3.4.1	DIAGRAMA DE SECUENCIA DE REGISTRAR USUARIO .....	57
3.4.2	DIAGRAMA DE SECUENCIA DE IDENTIFICAR USUARIO .....	58
3.4.3	DIAGRAMA DE SECUENCIA DE VER KEJAMAP.....	59
3.4.4	DIAGRAMA DE SECUENCIA DE REALIZARQUEJA .....	60
3.4.5	DIAGRAMA DE SECUENCIA DE CONSULTAR QUEJAS, APOYAR QUEJA Y REALIZAR COMENTARIO.....	61
CAPÍTULO 4.	IMPLEMENTACIÓN .....	62
4.1	SERVIDOR .....	62
4.1.1	SISTEMA DE GESTIÓN DE BASE DE DATOS .....	62
4.1.2	PHP .....	66
4.2	APLICACIÓN ANDROID .....	68
4.2.1	INTRODUCCIÓN .....	68
4.2.2	ESTRUCTURA DE LA APLICACIÓN .....	70
4.3	GOOGLE MAPS .....	83
4.4	PICASA WEB SERVICE .....	84
CAPÍTULO 5.	EXPERIMENTAL.....	85
5.1	INTRODUCCIÓN.....	85
5.2	PRUEBA DE RENDIMIENTO.....	85
5.2.1	PRUEBA DE RENDIMIENTO PARA REALIZAR QUEJA.....	85
5.3	PRUEBAS DE INTRODUCCIÓN DE DATOS .....	88
5.3.1	PRUEBAS DE INTRODUCCIÓN DE DATOS (DESBORDAMIENTO).....	88
5.3.2	PRUEBAS DE VALORES NULOS.....	89
5.4	PRUEBAS DE INTERGRACIÓN CON PICASA.....	93
5.4.1	PRUEBA DE DESBORDAMIENTO .....	93
5.4.2	PRUEBA DE ACCESO A IMÁGENES.....	93
5.4.3	PRUEBAS DE ACCESO CON DISTINTAS CUENTAS .....	94
CAPÍTULO 6.	ORGANIZACIÓN DEL PROYECTO.....	95
6.1	ORGANIZACIÓN Y COMPETENCIAS DEL PERSONAL .....	95
6.1.1	JEFE DE PROYECTO .....	95
6.1.2	ANALISTA.....	95
6.1.3	PROGRAMADOR.....	96





6.2	PLANIFICACIÓN DEL PROYECTO .....	96
6.2.1	TABLA DE TAREAS.....	96
6.2.2	DIAGRAMA DE GANTT.....	97
6.3	PRESUPUESTO .....	98
6.3.1	COSTE DEL PERSONAL .....	98
6.3.2	COSTE DEL MATERIAL UTILIZADO .....	98
6.3.3	PRESUPUESTO FINAL.....	99
CAPÍTULO 7.	FUTURAS LINEAS DE TRABAJO Y CONCLUSIONES .....	100
7.1	FUTURAS LINEAS DE TRABAJO .....	100
7.1.1	CREACIÓN DEL PORTAL WEB.....	100
7.1.2	ACUERDOS CON MEDIOS DE COMUNICACIÓN Y FUENTE RSS.....	100
7.1.3	COLABORACIÓN CON LAS ADMINISTRACIONES .....	101
7.1.4	SISTEMA DE RECOMPENSAS PARA EL USUARIO .....	101
CAPÍTULO 8.	REFERENCIAS Y BIBLIOGRAFÍA .....	103
CAPÍTULO 9.	APÉNDICES.....	106
9.1	HERRAMIENTAS Y TECNOLOGIAS UTILIZADAS.....	106
9.1.1	MYSQL .....	106
9.1.2	SERVIDOR APACHE .....	106
9.1.3	MOWES PORTABLE.....	106
9.1.4	GOOGLE MAPS .....	107
9.1.5	PICASSA WEB SERVICE.....	107
9.1.6	PHP .....	107
9.1.7	ECLIPSE .....	107
9.1.8	SAMSUNG GALAXY S .....	114
9.1.9	SAMSUNG GALAXY ACE.....	115
9.1.10	MICROSOFT OFFICE .....	115
9.1.11	MICROSOFT PROJECT .....	115
9.1.12	MICROSOFT VISIO.....	115



## FIGURAS:

Ilustración 1. Estructura del sistema .....	18
Ilustración 2. Fotodenuncia 1 .....	21
Ilustración 3. Fotodenuncia 2 .....	21
Ilustración 4. Fotodenuncia 3 .....	22
Ilustración 5. Fotodenuncias 1 .....	23
Ilustración 6. Fotodenuncias 2 .....	24
Ilustración 7. Fotodenuncias 3 .....	25
Ilustración 8. Fotodenuncias 5 .....	25
Ilustración 9. Fotodenuncias 4 .....	25
Ilustración 10. PonleFreno 1 .....	27
Ilustración 11. PonleFreno 2 .....	28
Ilustración 12. PonleFreno 3 .....	29
Ilustración 13. PonleFreno 4 .....	30
Ilustración 14. Diagrama de clases de la base de datos.....	55
Ilustración 15. Diagrama de clases de la aplicación Android .....	56
Ilustración 16. Diagrama de secuencia de registrar usuario .....	57
Ilustración 17. Diagrama de secuencia de identificar usuario .....	58
Ilustración 18. Diagrama de secuencia de verquejamap.....	59
Ilustración 19. Diagrama de secuencia de realizarqueja.....	60
Ilustración 20. Diagrama de secuencia de consultarqueja, comentarqueja y apoyarqueja .....	61
Ilustración 21. BBDD Tabla Usuarios .....	63
Ilustración 22. BBDD Tabla Quejas .....	64
Ilustración 23. BBDD Tabla Apoyos .....	65
Ilustración 24. BBDD Tabla Comentarios .....	65
Ilustración 25. Código PHP .....	66
Ilustración 26. Capas Android .....	68
Ilustración 27. Estructura Aplicacion Android.....	70
Ilustración 28. Pantalla Kejatactivity .....	71
Ilustración 29. Pantalla Registrar usuario.....	72
Ilustración 30. Pantalla Seleccionar opción.....	74
Ilustración 31. Pantalla Consultar queja.....	75
Ilustración 32. Pantalla Insertar queja .....	76
Ilustración 33. Pantalla Mostrar consulta .....	77
Ilustración 34. Pantalla Realizar queja .....	79
Ilustración 35. Pantalla Pintar mapa .....	81
Ilustración 36. APIS de Android.....	83
Ilustración 37. Código con Apikey de Google Maps.....	84
Ilustración 38. Gráfico Samsung Galaxy Ace .....	86
Ilustración 39. Gráfico Samsung Galaxy S .....	87
Ilustración 40. Tamaño maximo de campos .....	88
Ilustración 41. Pantallas de Introducción de datos .....	89



Ilustración 42. Caso Consultar queja .....	92
Ilustración 43. Diagrama de Gantt .....	97
Ilustración 44. ADT Android 1 .....	109
Ilustración 45. ADT Android 2 .....	110
Ilustración 46. ADT Android 3 .....	111
Ilustración 47. ApiKey GoogleMaps 1 .....	112
Ilustración 48. ApiKey GoogleMaps 2 .....	113
Ilustración 49. ApiKey GoogleMaps 3 .....	113



## TABLAS:

Tabla 1. Estadísticas de foros en España.....	15
Tabla 2. Comparativa aplicaciones.....	31
Tabla 3. Cuota de mercado de sistemas operativos .....	33
Tabla 4. Diagrama de Casos de Uso .....	41
Tabla 5. Caso de Uso 001: Registro de usuario .....	43
Tabla 6. Caso de Uso 002: Identificar usuario .....	43
Tabla 7. Caso de Uso 003: Realizar queja.....	43
Tabla 8. Caso de Uso 004: Consultar queja .....	44
Tabla 9. Caso de Uso 005: Ver Kejamap.....	44
Tabla 10. Caso de Uso 006: Apoyar queja.....	45
Tabla 11. Caso de Uso 007: Comentar queja .....	45
Tabla 12. Caso de Uso 008: Compartir queja .....	46
Tabla 13. RFN-001: Registro de Usuarios .....	47
Tabla 14. RFN-002: Identificación de Usuarios .....	47
Tabla 15. RFN-003: Registro de queja .....	48
Tabla 16. RFN-004: Selección de imagen .....	48
Tabla 17. RFN-005:Selección de dirección .....	48
Tabla 18. RFN-006:Consulta de quejas.....	49
Tabla 19. RFN-007:Vista de la lista de quejas .....	49
Tabla 20. RFN-008:Vista de queja seleccionada.....	50
Tabla 21. RFN-009: Apoyo de queja .....	50
Tabla 22. RFN-010: Comentario de queja .....	50
Tabla 23. RUS-001 .....	51
Tabla 24. RUS-002 .....	51
Tabla 25. RUS-003 .....	52
Tabla 26. RUS-004 .....	52
Tabla 27. RUS-005 .....	53
Tabla 28. RUS-006 .....	53
Tabla 29. RAP-001 .....	53
Tabla 30. RRE-001.....	54
Tabla 31. RSG-001 .....	54
Tabla 32. RLE-001 .....	55
Tabla 33. BBDD Tabla Usuarios .....	63
Tabla 34. BBDD Tabla Quejas .....	64
Tabla 35. BBDD Tabla Apoyos .....	65
Tabla 36. BBDD Tabla Comentarios.....	65
Tabla 37. Descripción PHPs .....	67
Tabla 38. Tiempos Samsung Galaxy S .....	86
Tabla 39. Tiempos Samsung Galaxy Ace .....	86
Tabla 40. Tareas del Proyecto .....	97
Tabla 41. Coste del Personal .....	98



Tabla 42. Coste del Material .....	98
Tabla 43. Presupuesto Final .....	99



# CAPÍTULO 1. DESCRIPCIÓN DEL PROYECTO

## 1.1 INTRODUCCIÓN

Este capítulo se centrará en dar una visión general del proyecto, describiendo el entorno en el que se ha desarrollado, su estructura y la materia a tratar en cada uno de los capítulos de este documento.

### 1.1.1 LA WEB 2.0

El término Web 2.0 fue acuñado por O'Reilly Media para una conferencia en 2004 donde intentarían explicar los cambios que estaba sufriendo la web. Si bien el término induce a pensar que el cambio sería una actualización del software o de la estructura física de la web, a lo que este concepto se intentaba referir era a los cambios que se estaban produciendo en cuanto al modelo de negocio y el comportamiento de los usuarios en la web.

Para entender estos cambios hay que partir de las clásicas páginas de HTML de solo lectura donde el usuario era un simple consumidor de contenido (a esto se le denominaría Web 1.0). Hoy en día el usuario no es solo consumidor de contenido sino que también es creador, comparte contenidos e interacciona con otros usuarios adquiriendo así una nueva dimensión social y participativa.

Si se tienen en cuenta los precios cada vez más bajos del hardware, el aumento de las conexiones inalámbricas y la masificación de los teléfonos móviles con acceso a la red junto con blogs, wikis, foros, grupos, redes sociales, etc., se puede ver cómo muchos usuarios se convierten, por ejemplo, en improvisados periodistas, críticos de cine o expertos en moda con cierto renombre y gran cantidad de seguidores.

## 1.1.2 FOROS, BLOGS, WIKIS, REDES SOCIALES

### FOROS

Los foros en Internet son también conocidos como foros de mensajes, de opinión o foros de discusión y son una aplicación web que le da soporte a discusiones u opiniones en línea.

Al contrario que las wikis en los foros no es posible modificar los aportes de otros miembros a no ser que se tengan cuentas especiales de administrador. Hoy en día en día los foros no solo se dedican a compartir opiniones sino que mediante el auge de los servidores de almacenamiento online se han aprovechado también para compartir archivos mediante sus *url's*, añadiendo mas funcionalidades y aumentando así el número de integrantes. Es tal la magnitud de las opiniones vertidas en los foros que en un ranking realizado en España en Marzo de 2012 los mensajes se cuentan por millones.

FORO	MENSAJES
ForoCoches	105M (500.000 nuevos mensajes semanales)
MeriStation	32M (90.000 nuevos mensajes semanales)
ElOtroLado	24M (70.000 nuevos mensajes semanales)

Tabla 1. Estadísticas de foros en España

### BLOGS

Un blog es un sitio web periódicamente actualizado que recopila cronológicamente textos o artículos de uno o varios autores, apareciendo primero el más reciente, donde el autor conserva siempre la libertad de dejar publicado lo que crea pertinente. Además la gente puede realizar comentarios sobre esos artículos estableciendo un *feedback* con el autor.

### WIKIS

Es un sitio web colaborativo que puede ser editado por varios usuarios. Los usuarios de una wiki pueden así crear, editar, borrar o modificar el contenido de una página web, de forma interactiva, fácil y rápida. El fin que persiguen las wikis es la creación de un conocimiento común creado por y para los usuarios. La wiki más importante actualmente es Wikipedia , la cual consta de más de 20 millones de artículos en 282 idiomas y que forzó a Microsoft en 2009 a dejar de vender su enciclopedia Encarta debido a las bajas ventas producidas por incremento del uso de esta gigantesca wiki.

### REDES SOCIALES

Una red social es una estructura social que se puede representar en forma de uno o varios grafos en el cual los nodos representan individuos (a veces denominados actores) y las aristas relaciones entre ellos. Las relaciones pueden ser de distinto tipo, como intercambios



financieros, amistad, relaciones sexuales, o rutas aéreas. En España según un estudio realizado en Marzo de 2012 el 68,4 % de los usuarios acceden diariamente a redes sociales. De las cuales los porcentajes de uso son:

- Facebook es utilizada por el 89,8% de los usuarios de redes sociales.
- Twitter: 36,9%.
- Google+: 25,5%.
- Tuenti: 25,4%.
- LinkedIn: 16,9%.
- Flickr: 7,4%.
- Badoo: 5,6%.
- Foursquare: 4,2%

### CONCLUSION

Tras el análisis de las distintas partes de la web 2.0 se puede concluir que cada una de ellas tiene distintos fines. Los foros persiguen la participación, las wikis una colaboración para crear un conocimiento común, los blogs persiguen la reacción y las redes sociales se centran en el contacto entre personas, pero todos ellos tienen una gran repercusión social.

#### 1.1.3 APLICACIONES DE COLABORACIÓN CIUDADANA

Hoy en día la web 2.0 ha evolucionado. Al tener una gran capacidad de comunicación, se ha empezado a utilizar en los ámbitos de colaboración ciudadana, tal y como hace hoy en día la página web de la policía, otras plataformas como ponle freno del Grupo Planeta o la aplicación que se va a desarrollar en este proyecto.

Con ello se ve como la informática está cada vez más ligada a las necesidades de los usuarios, y si bien anteriormente protestas o denuncias se tenían que hacer personalmente o a través de comunidades de vecinos o plataformas sociales, hoy en día se puede hacer desde cualquier lugar accediendo a Internet.

### 1.2 OBJETIVOS

El objetivo principal de este proyecto es la implementación de un sistema de colaboración ciudadana que permita a los usuarios realizar quejas sobre desperfectos de su entorno aprovechando la portabilidad y otros recursos que proporcionan los *smartphones*.





Esta iniciativa no sólo permite que el usuario pueda realizar quejas sobre su entorno, sino también informar sobre posibles desperfectos que pueden resultar peligrosos para cualquier persona que pase por una determinada zona. Además de esto, la iniciativa pretende ser una fuente de información periodística y una medida de presión para que los distintos ayuntamientos mejoren el mantenimiento de sus localidades.

El objetivo secundario de este proyecto será investigar qué nos ofrece el sistema operativo que se utilizará en este proyecto (Android) y las distintas opciones que nos ofrece su API para intentar mejorar nuestra aplicación.

A parte de lo anteriormente explicado se investigará sobre cómo crear un servidor y una base de datos que se adecuen a las necesidades del proyecto y a cómo utilizar servicios Web como por ejemplo Picasa.

### 1.3 MEDIOS UTILIZADOS

Para el desarrollo de este proyecto se han utilizado diversos medios que se agruparán en tres bloques: servidor, aplicación Android y servicios web.

#### 1.3.1 SERVIDOR

Para la parte del servidor se ha utilizado un ordenador portátil HP Pavlion dv6 con un microprocesador Dual-Core de 2,00 GHz. Sobre él se ha instalado un aplicación llamada Mowes Portable la cual contiene un servidor Apache, un Sistema gestor de BBDD MySQL y PHP 5.0. En el será guardará toda la información correspondiente a las quejas realizadas.

#### 1.3.2 APLICACIÓN ANDROID

Para el desarrollo de la aplicación se ha utilizado el entorno Eclipse con un Plugin específico para Android, además para realizar las pruebas y ejecutar la aplicación se ha utilizado dos móviles Samsung, modelos Galaxy S y Galaxy Ace.

#### 1.3.3 SERVICIOS WEB

Para el desarrollo del proyecto ha sido necesaria la utilización de dos servicios web. Google Maps para proveer al sistema de mapas donde geoposicionar las quejas, y el servicio web de Picasa para poder almacenar las fotos de las quejas realizadas.

## 1.4 ESTRUCTURA DE LA APLICACIÓN

El diagrama que se muestra a continuación proporciona una idea del sistema que se va a desarrollar en este proyecto.

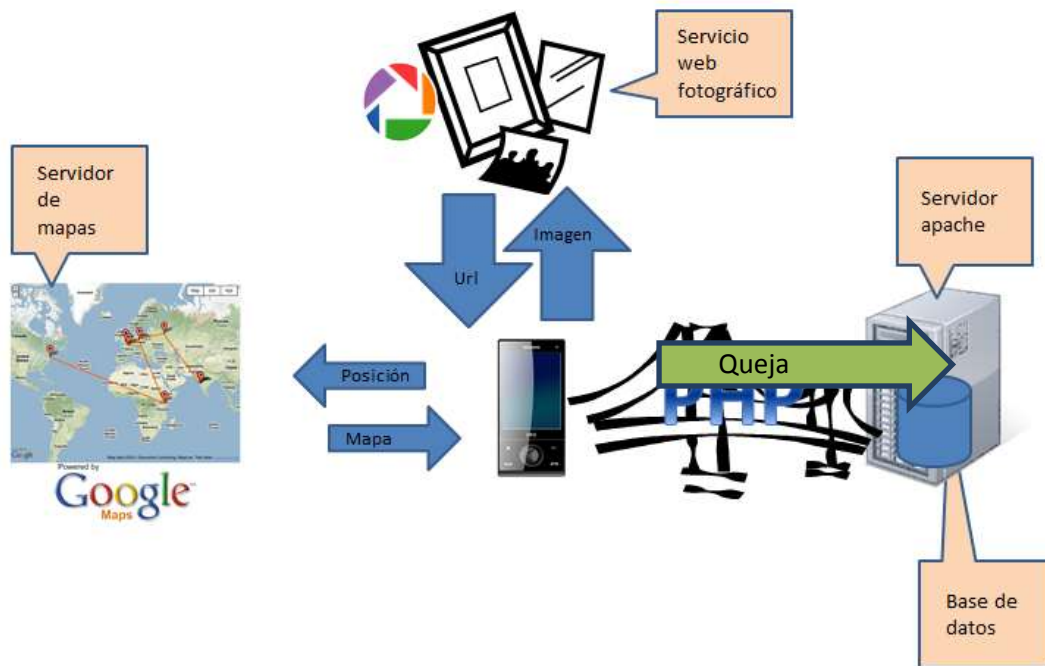


Ilustración 1. Estructura del sistema

Como se puede apreciar en el esquema, el usuario desde su terminal introducirá una queja con su correspondiente imagen y posición. La imagen se envía a un servicio web fotográfico donde se almacenará. Una vez almacenada se proporcionará una *url* para poder acceder a ella. Esta *url* junto con los demás datos formará una queja, que a través de un PHP se introducirá en una base de datos MySQL alojada en un servidor Apache.

Por otro lado, si se quiere obtener las quejas de una determinada zona se realizará una consulta a la base de datos a través de un PHP, y se obtendrá un mapa del servidor de mapas a partir de la posición del GPS. Una vez se tiene todo esto se pinta una capa sobre el mapa con las posiciones de todas las quejas obtenidas.



## 1.5 ESTRUCTURA DEL DOCUMENTO

Este documento consta de 9 capítulos:

- Capítulo 1: Es el capítulo actual, sirve de introducción para que el lector pueda tener una ligera idea del proyecto que se va a desarrollar.
- Capítulo 2: En este capítulo se analizarán las aplicaciones similares y las distintas opciones para poder desarrollar el proyecto.
- Capítulo 3: En él se centrarán las tareas de análisis y diseño del proyecto. Mostrando los casos de uso, la especificación de requisitos y una serie de diagramas que darán una idea de cómo se ha de realizar la aplicación.
- Capítulo 4: En este capítulo se describirá como se ha realizado la aplicación analizando sus distintas partes.
- Capítulo 5: En él se mostrarán todas las pruebas realizadas a la aplicación así como los resultados obtenidos.
- Capítulo 6: En este capítulo se mostrará la planificación del proyecto y los costes que conllevaría si se quisiera desarrollar.
- Capítulo 7: En él se describen las posibles líneas futuras de trabajo en caso de que el proyecto tuviera éxito.
- Capítulo 8: En él se muestran las fuentes consultadas para el desarrollo del proyecto.
- Capítulo 9: Contiene información complementaria sobre los medios utilizados en el proyecto.



## CAPÍTULO 2. ESTADO DEL ARTE

### 2.1 INTRODUCCIÓN

En este capítulo se mostrará el proceso de análisis en investigación, el cuál ayudará a definir mejor las metas del proyecto y las posibles soluciones para conseguirlas. Dicho esto, se pueden apreciar en este capítulo dos partes:

Por un lado se estudiarán las distintas aplicaciones móviles que se encuentran ya desarrolladas y se analizará si cubren todas las necesidades del cliente y cuáles son sus puntos fuertes y sus puntos débiles. Por el otro, se analizarán las distintas soluciones para desarrollar el proyecto e intentar elegir las más óptimas.

### 2.2 ESTUDIO DE APLICACIONES SIMILARES

En este apartado se analizarán las distintas aplicaciones que hay desarrolladas y que intentan cumplir una función similar a la de este proyecto, entre ellas se encontrarán aplicaciones móviles y páginas web; unas son accesibles desde todas las plataformas y otras solo están disponibles para alguna de ellas. Con ello no sólo se pretende desmarcarse en el mercado nacional, sino crear una aplicación de calidad a partir de los puntos fuertes de las analizadas.

#### 2.2.1 FOTODENUNCIA.ORG

Quizás la aplicación más completa en cuanto a las que se encuentran en el mercado nacional.

Esta aplicación se encuentra disponible para Iphone en la AppStore y además dispone de un portal web que permite consultar las denuncias sobre un mapa [16].

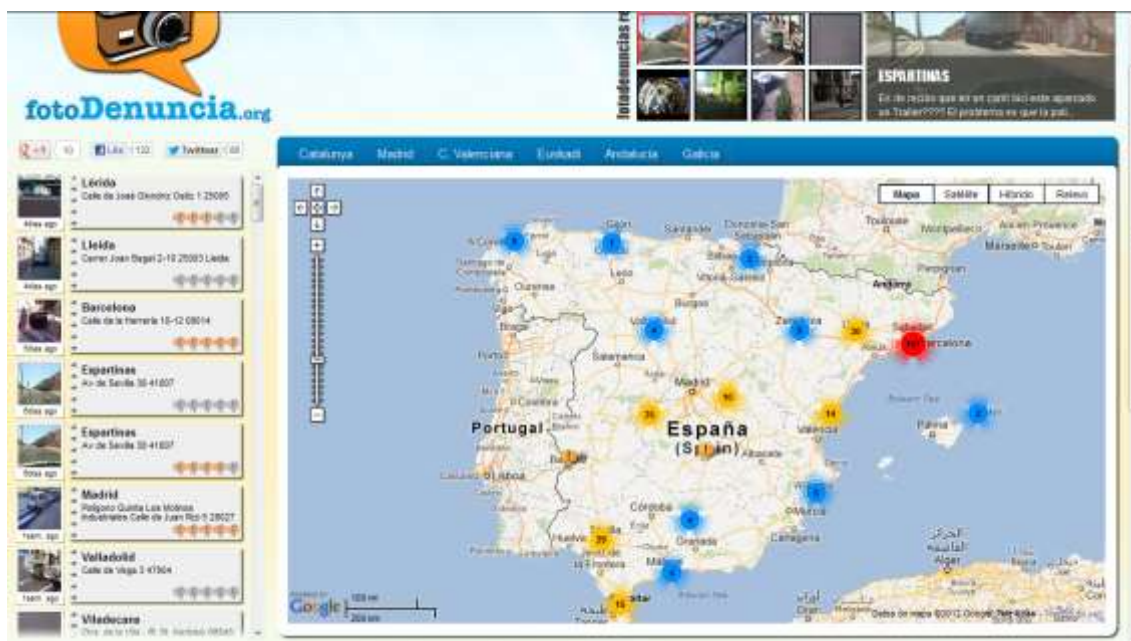


Ilustración 2. Fotodenuncia 1

Esta es una imagen del portal de fotodenuncia.org. En primer lugar se observa cómo en el mapa se encuentran las denuncias organizadas geográficamente, pero no permite introducir ninguna dirección para ir directamente a ella, se tendría que navegar sobre el mapa haciendo zoom para poder acercarse a la zona seleccionada.

En la parte de la izquierda se encuentra una lista de las últimas denuncias ordenadas por fecha, pinchando sobre ellas aparecerá el mapa centrado en ella y pinchando sobre su localización en el mapa ampliará su información.

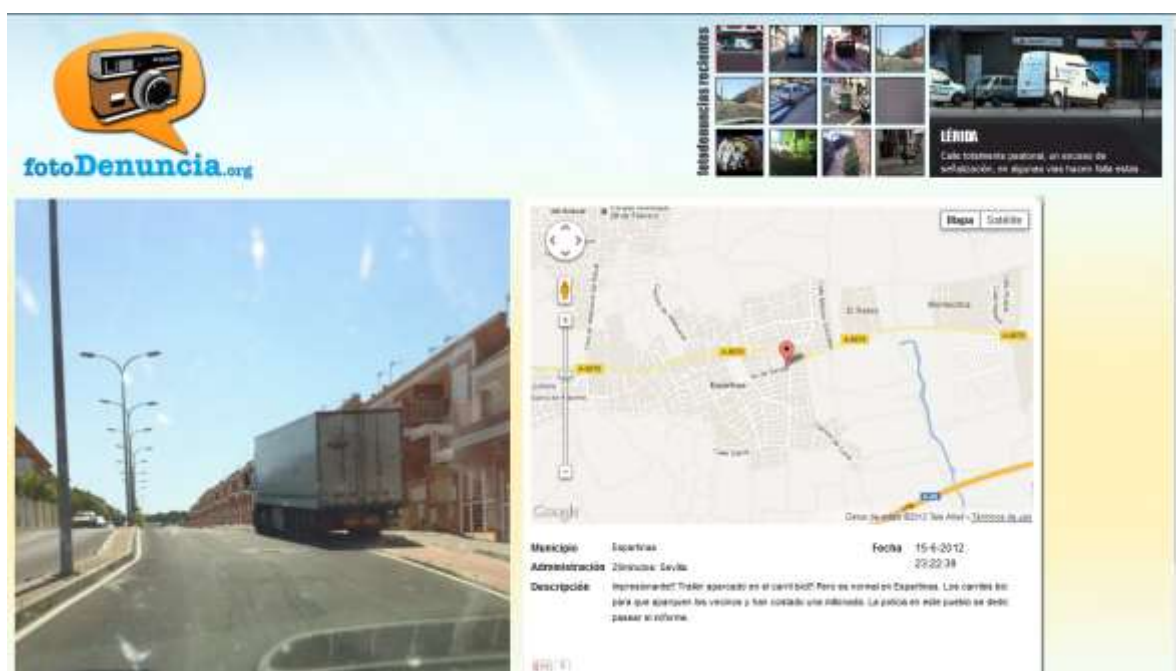


Ilustración 3. Fotodenuncia 2

Como se puede ver, la información ampliada consta del municipio, administración, descripción y fecha, y mirando un poco más abajo se puede compartir la denuncia en Google+, una red social que aunque está en auge, no es de las más populares como pueden ser Facebook o twitter.

Como se ha podido observar es una aplicación bastante completa, con algunos detalles que se podrían mejorar, pero en general está bien desarrollada.

A continuación se va a analizar el punto fuerte de esta aplicación, el cual no está disponible en el portal pero sí en la aplicación móvil.



Ilustración 4. Fotodenuncia 3

Al seleccionar la opción “hacer una fotodenuncia” da la posibilidad de mandar un correo a la Administración correspondiente para que quede constancia de ella, sin duda una buena idea y quizás por lo que esta aplicación sea una de las más completas.

Si se analiza cómo se obtiene la información de las administraciones, se puede ver en la imagen como hay una opción en la que se puede dar de alta una Administración, la cuál recibirá las denuncias de una determinada zona. Al no haber en la aplicación ningún tipo de identificación del usuario que realiza la denuncia o que da de alta una Administración, no se puede asegurar que la información sea veraz y por lo tanto se puede tener un problema de seguridad. Es decir, que una persona se registre como una administración y no sea tal y le llegue información personal como el número de móvil o el correo de determinados

denunciantes de la zona que haya registrado. Es más, en la primera imagen se puede observar cómo la administración es 20 minutos-Sevilla y no una administración pública.

Por lo demás es una gran aplicación y cumple de manera bastante completa las funciones que pretende proporcionar.

Para más información hay un video explicativo de su utilización en la web, el cual se adjunta en el siguiente link [http://www.youtube.com/watch?feature=player\\_embedded&v=9BzuCH\\_HwtM](http://www.youtube.com/watch?feature=player_embedded&v=9BzuCH_HwtM).

### 2.2.2 FOTODENUNCIAS.NET

Fotodenuncias.net al igual que fotodenuncia.org, consta de un portal y una aplicación, la cual está disponible tanto para Iphone como para Android en sus correspondientes tiendas virtuales.

En la web de Fotodenuncias.net se puede realizar tanto una denuncia, como consultar denuncias y hacer comentarios sobre ellas, sin embargo, por el móvil sólo se pueden enviar las denuncias y por la web sólo se pueden realizar consultas y comentarios sobre las denuncias [17].



### Ilustración 5. Fotodenuncias 1



La ilustración 5 muestra la página de inicio de la web fotodenuncias.net. Como se puede observar la página contiene publicidad entre los elementos de esta, perjudicando así la navegación del usuario y creando una experiencia menos agradable. Para explicar en qué consiste la página, lo explica en un par de párrafos con una maquetación bastante pobre y un tamaño y tipo de letra que poco se diferencian de los mensajes que muestran los navegadores sobre información de una determinada página al no encontrarse algún tipo de contenido. Pero lo que más puede perjudicar la experiencia del usuario en la navegación de esta página es la falta de diferenciación entre las distintas partes, como por ejemplo que el botón de examinar se encuentra al principio de la página incrustado en un texto y una imagen con fondo de igual color que la página y que no muestra ningún tipo de cambio cuando se pasa el puntero o se pincha sobre él. Esto puede llevar a pensar al usuario que se puede tratar de un banner publicitario y que no lo pinche la primera vez que lo vea, haciendo así la navegación más lenta y menos intuitiva.

Si se pincha en cualquiera de las denuncias se puede observar cuatro partes: una foto de la denuncia con su correspondiente título, fecha y la posibilidad de poder compartir la denuncia en Facebook y Twitter (ilustración 6 ), un mapa que muestra dónde está situada la denuncia (si es que se ha proporcionado esta información)(imagen1), un banner publicitario, y por último y más interesante, los comentarios que ha realizado la gente sobre la denuncia y la posibilidad de que el usuario pueda realizar uno.



Ilustración 6. Fotodenuncias 2



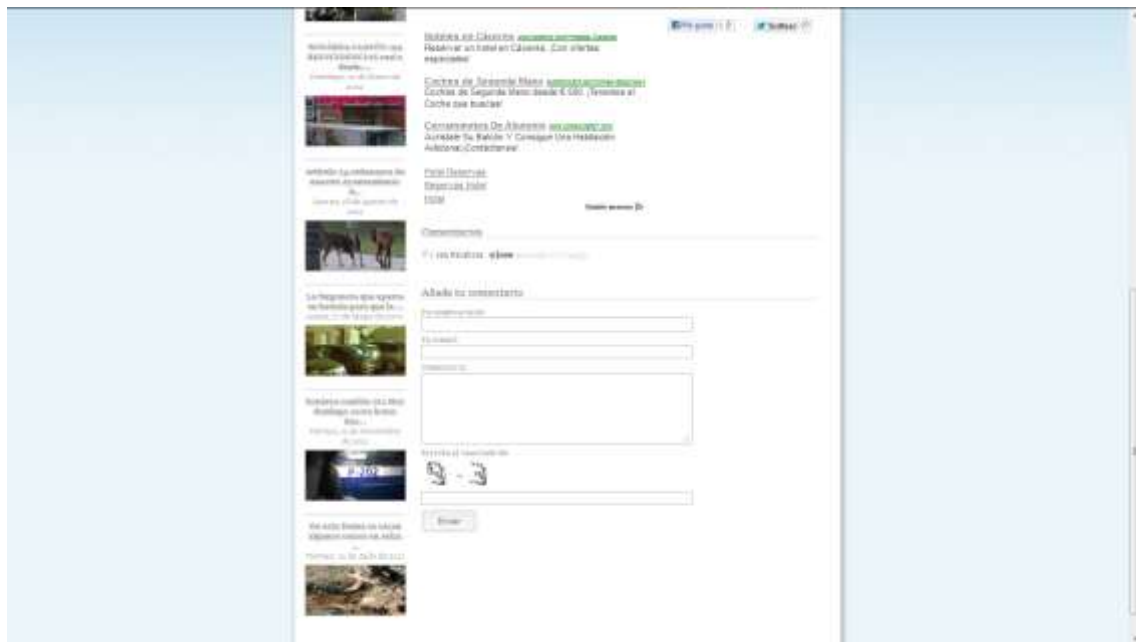


Ilustración 7. Fotodenuncias 3

A continuación se pasará a analizar las aplicaciones de fotodenuncias.net tanto para Iphone como para Android.



Ilustración 9. Fotodenuncias 4



Ilustración 8. Fotodenuncias 5

Como se puede observar tanto en la versión para Iphone(imagen de la izquierda), como en la versión para Android(imagen de la derecha), únicamente se permite realizar una denuncia, en ningún momento consultar las ya publicadas. Además en la aplicación para Android no permite realizar una fotografía con la cámara y únicamente permite subir una imagen si está alojada en el teléfono, pero no sólo eso, esta aplicación accede a al GPS para



buscar la posición y asociarla a la denuncia, limitando únicamente a poder subir la denuncia desde el lugar en el realizado la fotografía.

En conclusión fotodenuncias.net es un servicio en conjunto peor que fotodenuncia.org pero tiene una funcionalidad interesante que es la posibilidad de añadir comentarios, la cual, permite que la denuncia tenga mayor repercusión social y proporciona al denunciante un mayor reconocimiento.

### 2.2.3 PONLE FRENO

Ponle Freno es una plataforma ciudadana creada por el grupo *Planeta* (*Antena 3, Europa FM, Marca...*) en la que se pueden denunciar y consultar desperfectos en las distintas carreteras y calzadas de toda España [18].



The screenshot displays the Ponle Freno website interface. At the top, there is a navigation bar with the Ponle Freno logo, a '¡Firmar!' button, and a menu with links: INICIO, RADARES, NOTICIAS, PLATAFORMA CIUDADANA, DENUNCIAS, CARRERA PONLE FRENO 2011, CONSEJOS, JORNADAS 3.0, PREMIOS, and CAMPAÑAS. Below the navigation bar is a large banner featuring a road with a 'PONLE FRENO' stop sign and the text 'ENVÍA TUS PROPUESTAS'. To the right of the banner, a 'PROPUESTA: Plataforma Ciudadana' section explains the platform's goal to reduce traffic accidents and improve road safety.

Below the banner, the main content area is titled '¡JUNTOS SI PODEMOS!' and 'Únete a la Plataforma Ciudadana Ponle Freno y ayuda a salvar vidas'. It states that over 65,000 people are already part of the platform and encourages users to join to help save lives. A section titled 'ESPERAMOS TUS PROPUESTAS' explains that users can submit proposals by filling out a form. Below this, a login section titled 'Entra en Ponle Freno' prompts users to enter their DNI and password to access the platform. A 'CONECTAR' button is provided for login. Links for 'He olvidado la contraseña', 'Modifica tus datos', and 'Si quieres darte de baja, Pulsa aquí' are also visible.

On the right side of the page, there is a 'PLATAFORMA CIUDADANA' section showing a counter for 'Somos 90716' members. Below this, there are social media links for Twitter and Facebook, and a list of people who like the page. At the bottom right, there is a 'Lo último' section with news items about the platform's success in getting the administration to move, a campaign for drivers to be more attentive to rearview mirrors and motorcycles, and a reminder to maintain a safe distance from motorcycles.

Ilustración 10. PonleFreno 1

Como se puede observar en la imagen superior, el usuario para poder realizar una denuncia tiene que registrarse y para ello debe rellenar el siguiente formulario:



¡Rellena los siguientes datos y apúntate!

**PONLE FRENO**

Cerrar ventana X

Nombre y apellidos

D.N.I./N.I.E

e-mail

Contraseña

Confirma tu contraseña

Fecha de nacimiento  -  -

Provincia

sexo: ☐ Hombre ☐ Mujer

carnet de conducir: ☐ Si ☐ No

Acepto las condiciones ☐

**Apúntate**

Ilustración 11. PonleFreno 2

Cuando se analiza el formulario se puede ver que se reclaman gran cantidad de datos de carácter personal del usuario, los cuales si se leen las condiciones, pasan a incluirse en una base de datos cuyo responsable es Antena 3 Televisión.

Por otro lado si se quiere consultar las denuncias ya realizadas, se puede acceder a ellas libremente sin registrarse en ningún lado, pero cuando se accede a “todas las denuncias” se observa lo siguiente:



Ilustración 12. PonleFreno 3

Un listado de denuncias sin ningún tipo de organización ni por posición geográfica, ni por fecha, ni por número de visitas, etc.



Si se selecciona alguna de ellas se puede ver cómo se amplía la información de manera escasa, mostrando una descripción y permitiendo hacer comentarios sobre ella en la propia página y en gran cantidad de redes sociales.

Ponle Freno Denuncias Todas las denuncias

## Caída sobre un carril bici

Mario Carlos Fraile Vergel:  
Avenida de Carlos III, número 46 de Córdoba.

Me gusta 0 Twitter 0 Compartir 0 Comentarios



Avenida de Carlos III, número 46 de Córdoba

Avenida de Carlos III, número 46 de Córdoba | Foto: Ponle Freno

Mario Carlos Fraile Vergel | Avenida de Carlos III, número 46 de Córdoba. | Actualizado el 30/03/2011 a las 16:14 horas

Está caída sobre un carril bici que además interrumpe el paso de las bicicletas porque hay un coche atravesado en el mismo. Esta señal está situada en la Avenida de Carlos III, número 46 de Córdoba.

Me gusta 0 Twitter 0 Compartir 0 Comentarios



Compartir en: ☐ Facebook ☐ Windows Live

COMENTAR

Ilustración 13. PonleFreno 4

Con ésta última imagen se puede concluir que aunque este portal exige gran cantidad de datos personales y no tiene ningún tipo de organización en cuanto a mostrar el listado de las denuncias, lo que sí que posee es una gran repercusión social y mediática, tanto en las redes sociales como en la propia cadena de televisión que lo ha creado.

## 2.2.4 CONCLUSIÓN

A continuación se mostrará una tabla comparativa con todas las aplicaciones y portales analizados, para intentar averiguar cuáles son las funcionalidades apropiadas para que la futura aplicación que se realice, sea de calidad y pueda competir con las anteriormente analizadas.

Funcionalidades	Fotodenuncia.org	Fotodenuncias.net	Ponle Freno
Información correctamente ordenada			
Geolocalización de la denuncia	A través de gps	A través gps	
Necesidad de registro para realizar denuncia			
Mapa de denuncias			
Soporte móvil			
Sincronización con redes sociales			
Soporte mediático			
Realizar comentarios sobre una denuncia			
Colaboración directa con instituciones	Solo las dadas de alta en la aplicación		Dgt

Tabla 2. Comparativa aplicaciones

Después de este análisis se puede tener una idea clara sobre las funcionalidades que son deseables en la aplicación a desarrollar:

- Mostrar la información de una manera organizada y clara.
- Tener todas las denuncias geolocalizadas de una manera correcta y sencilla.
- Mostrar un mapa con todas las situaciones de las denuncias.
- Tener la posibilidad de compartir las denuncias en redes sociales.
- Tener la posibilidad de poder realizar comentarios sobre una denuncia.
- Tener un respaldo mediático.
- Tener un respaldo institucional.



Las características de que la aplicación tenga un respaldo mediático e institucional son deseables pero no se podrán incluir posteriormente en el apartado de diseño, debido a que conllevarían la implicación de acuerdos tanto institucionales como comerciales a los cuales todavía no se ha llegado.

### 2.3 ESTUDIO DE LAS POSIBLES SOLUCIONES

Como se ha visto en el apartado anterior, a partir del análisis de las distintas aplicaciones se ha conseguido definir las características necesarias para que la aplicación a desarrollar sea de calidad.

El objetivo de este apartado es buscar las mejores soluciones para que estas características estén presentes en nuestra aplicación. Debido a que la elección de las distintas partes puede influir en las demás, se intentará buscar antes una solución a las partes más influyentes y a continuación, a las que tengan menos relevancia en las demás partes de la aplicación.

#### 2.3.1 PLATAFORMA DE DESARROLLO

En este apartado se analizará las distintas plataformas de desarrollo para la aplicación. Como el proyecto se realizará para dispositivos móviles se tendrá en cuenta distintas características como son la cuota de mercado en España y la compatibilidad con los distintos dispositivos.

##### *CUOTA DE MERCADO*

A continuación se pasará a mostrar una gráfica de un estudio realizado por la empresa Statcounter en el que se analiza la cuota de mercado de los distintos sistemas operativos para móviles en España desde Abril de 2011 hasta abril de 2012.



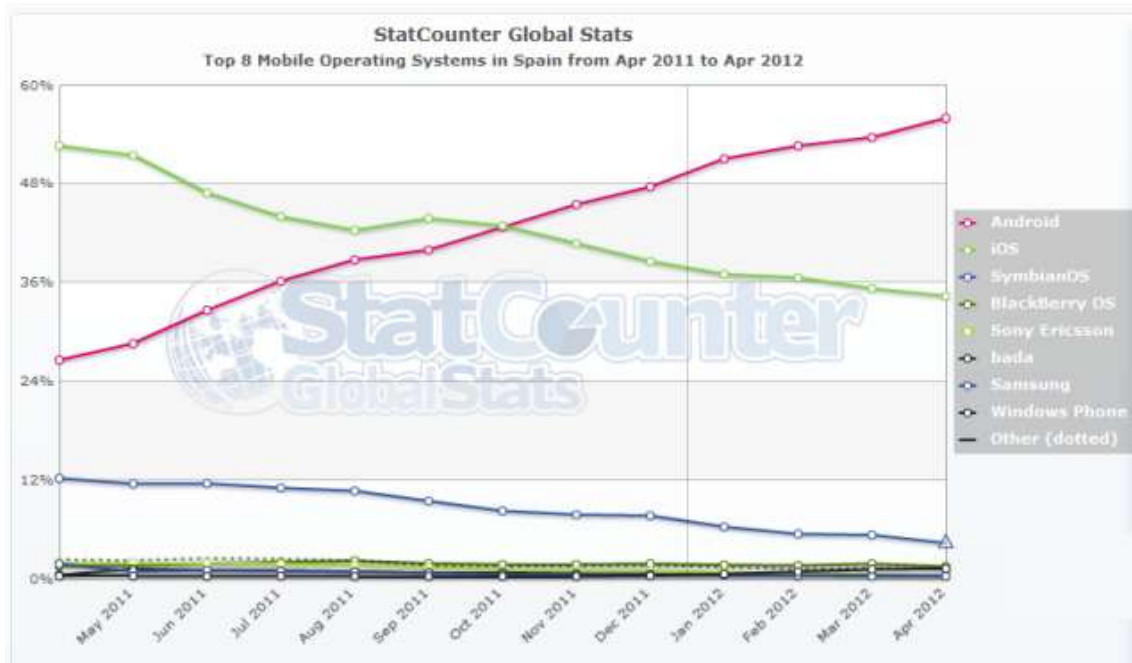


Tabla 3. Cuota de mercado de sistemas operativos

Como se puede observar los sistemas operativos predominantes son Android y iOS, los cuales juntos acaparan más del 90% de la cuota de mercado. También se puede deducir que mientras iOS ha ido descendiendo prácticamente desde Mayo de 2011, el crecimiento de Android ha sido constante desde esa misma fecha situándolo a partir de Octubre de 2011 en el primer puesto de los sistemas operativos móviles. Por tanto, se puede decir que en este sentido Android le gana la partida a iOS.

Visto lo anterior, se podría decir que Android ya tiene un punto a su favor para desarrollar la aplicación para su sistema operativo, pero aun así se compararán más profundamente estos dos sistemas operativos.

### COMPARATIVA ANDROID VS IOS

- **Lenguaje de programación:** Mientras que iOS utiliza Objective-C, un lenguaje prácticamente desarrollado para dispositivos Apple, Android utiliza Java para la parte funcional y XML para la parte gráfica, dos lenguajes mucho más conocidos.
- **Rendimiento:** Quizás uno de los pocos apartados en los que iOS esté por delante de Android. Esto es debido a que mientras Android utiliza una máquina virtual para ejecutar sus aplicaciones Java, iOS al estar sus aplicaciones programadas en Objective-C ejecuta sus aplicaciones directamente sin necesidad de intermediarios.
- **Entorno de desarrollo:** Aunque para desarrollar en Android se pueden utilizar distintos entornos, el que más facilidades da y el más usado es el IDE Eclipse con su correspondiente plugin. Sin embargo si se quiere desarrollar para iOS se utiliza el XCode, un IDE de MAC que únicamente se usa en ordenadores con el sistema operativo de Apple OSX, mientras que Eclipse se encuentra disponible para todos los sistemas operativos.

- **Precio de la licencia de desarrollador:** Para poder subir aplicaciones a sus correspondientes “Markets” en las dos plataformas se deberá conseguir una licencia de desarrollador y ninguna de ellas es gratuita. Por un lado está la licencia de desarrollador de Android, la cual tiene un coste de \$25 y es vitalicia; y por otro la licencia de desarrollador de iOS la cual tiene un coste de \$99 anuales, difícilmente amortizable si se desarrollan aplicaciones gratuitas.
- **Distribución:** Para distribuir las aplicaciones estos dos sistemas operativos utilizan sus propias tiendas virtuales, en el caso de Android “Google Play”, y en el caso de iOS “App Store”.

Para poder subir una aplicación a Google Play una vez terminada, únicamente se necesita identificarse con la correspondiente licencia, subir el APK correspondiente y en unas pocas horas estará disponible en la tienda. Sin embargo este no es el caso de la App Store en la cual si se quiere subir una aplicación, tendrá que pasar una serie de filtros, que puede tardar en torno a una semana.

### *HTML5 Y DESARROLLO MULTIPLATAFORMA VS APLICACIONES NATIVAS*

Otra solución que se presenta a la hora de desarrollar para dispositivos móviles es la del desarrollo de aplicaciones en HTML5, que es la revisión del lenguaje HTML y que añade muchas más opciones que su predecesor.

La principal ventaja de este HTML5 es que este lenguaje es interpretable por prácticamente la mayoría de los dispositivos, teniendo de esta manera que hacer un único desarrollo para un proyecto el cual será válido para todos los terminales.

Analizando esta solución probablemente sería una de las mejores candidatas para el desarrollo del proyecto porque como se ha dicho anteriormente uno de los objetivos de este proyecto, es hacer llegar la aplicación a la mayor cantidad de gente posible. Pero no todo son beneficios con HTML5 porque si se compara con las aplicaciones nativas de cualquier plataforma, estas últimas permiten proporcionar una mejor experiencia al usuario, ya sea por su mayor rendimiento o por la posibilidad del acceso al hardware del dispositivo mediante las bibliotecas de su API, además HTML5 se ejecuta sobre el navegador del dispositivo, y en Android (la mayoría de dispositivos móviles en España) por debajo del navegador no solo está el sistema operativo sino la máquina virtual java, estando de esta manera una aplicación de HTML5 demasiado alejada del hardware como para que tenga un buen rendimiento.

### *CONCLUSIÓN*

Como se puede observar en la comparativa y según los puntos analizados en la comparativa de Android vs iOS, Android se adapta mejor al desarrollo del proyecto, ya no sólo por los costes del desarrollo sino porque uno de los objetivos que persigue este proyecto es presentar la aplicación a la mayor cantidad de personas posibles, y según se ve en la cuota de mercado Android sale ganando en este punto. Pero si sólo se tuviera en cuenta ésta última



característica, desarrollar en HTML5 sería la mejor opción; no obstante, otra meta que persigue este proyecto es desarrollar una aplicación de calidad y que proporcione una buena experiencia al usuario y como se ha observado en la comparativa de HTML5 ésta es mejor con una aplicación nativa. Por lo tanto, después de todo lo analizado se puede concluir que la plataforma de desarrollo Android es la que más se ajusta a las necesidades de la aplicación.

### 2.3.2 BASE DE DATOS

Como se ha visto en el análisis de las anteriores aplicaciones se necesitará el soporte de algún tipo de servidor donde almacenar los datos de los usuarios, las denuncias, etc.

Dada la naturaleza de aplicación que es permitir divulgar y consultar la información sobre desperfectos en nuestra zona, no se trabajará con datos de carácter personal, ni se necesitará de una gran base de datos que tenga multitud de funcionalidades; por ello una de las mejores soluciones es una base de datos gratuita, versátil, liviana y que no sobrecargue demasiado el servidor. Esta es prácticamente la definición de MySQL, pero aun así se comparará con otro sistema de base de datos como Microsoft Access.

#### MYSQL

MySQL es un sistema de gestión de bases de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones. MySQL es un sistema de administración relacional de bases de datos. Una base de datos relacional archiva datos en tablas separadas en vez de colocar todos los datos en un gran archivo. Esto permite velocidad y flexibilidad. Las tablas están conectadas por relaciones definidas que hacen posible combinar datos de diferentes tablas sobre pedido.

MySQL es software de fuente abierta. Fuente abierta significa que es posible para cualquier persona usarlo y modificarlo. Cualquier persona puede bajar el código fuente de MySQL y usarlo sin pagar. Cualquier interesado puede estudiar el código fuente y ajustarlo a sus necesidades. MySQL usa el GPL (GNU General Public License) para definir qué puede hacer y qué no puede hacer con el software en diferentes situaciones.

#### ACCESS

Microsoft Access forma parte del paquete ofimático de Microsoft llamado Office, el cual no tiene una licencia libre y hay que comprar.

Por medio de Microsoft Access, puede administrar toda la información desde un único archivo de base de datos. Dentro del archivo, puede utilizar:

- Tablas para almacenar los datos.



- Consultas para buscar y recuperar únicamente los datos que necesita.
- Formularios para ver, agregar y actualizar los datos de las tablas.
- Informes para analizar o imprimir los datos con un diseño específico.
- Páginas de acceso a datos para ver, actualizar o analizar los datos de la base de datos desde Internet o desde una intranet
- Almacenar los datos una vez en una tabla y verlos desde varios lugares.

### COMPARATIVA MYSQL VS ACCESS

Esta comparativa se fijará principalmente en los siguientes aspectos:

- **Funcionamiento:** MS Access tiene un funcionamiento distinto al de la mayoría de SSGGBDD (Sistemas Gestores de Bases de Datos). Dispone de una interfaz de usuario (el propio programa) que está íntimamente ligada al (único) fichero de base de datos. Mientras que MySQL instala en el sistema operativo un proceso servidor gestionado por el propio sistema operativo y que es accesible desde el puerto de escucha. Esto se podría hacer con Access, pero para ello sería necesario crear una fuente de datos ODBC.
- **Rendimiento:** MySQL ofrece un rendimiento mucho mayor al de MS Access. Hay que tener en cuenta que hoy en día MySQL es uno de los SSGGBDD más rápidos, probablemente por su falta de funcionalidad ante sistemas más robustos como Oracle.
- **Capacidad:** MS Access comenzará a tener problemas a partir de un número crítico de registros, que dependerá de las características de la máquina, de las versiones de software, etc. En cualquier caso, MySQL soportará un número bastante mayor de registros.
- **Tipo de licencia:** MySQL es Open Source, lo que implica que tienes acceso incluso al código fuente, y que la licencia de uso es gratuita. MS Access es Microsoft, y hay que pagar por su uso, además de que el código fuente de la aplicación es sólo conocido por Microsoft.
- **Flexibilidad:** MySQL es multiplataforma. Funciona sobre Linux, Windows, Solaris, etc. MS Access sólo funciona sobre Windows.

### CONCLUSIÓN

Microsoft Access es una herramienta adecuada si se quiere crear una base de datos pequeña de contactos o de determinados clientes. Pero en el caso de este proyecto, que se trata de una base de datos al uso sobre un servidor apache, lo que más se ajusta es una base de datos MySQL.

### 2.3.3 CONEXIÓN DE LA APLICACIÓN CON LA BASE DE DATOS

Si bien ya se ha elegido la base de datos, ahora se tendrá que elegir una manera de conectar la aplicación con la base de datos para poder realizar consultas, inserciones, etc.

La plataforma de Android dispone de librerías para manejar bases de datos, e incluso como utiliza Java se tiene la posibilidad de importar librerías de Java apropiadas para ello. Sin embargo la solución que se escogerá para éste proyecto será mediante la utilización de PHP, ya no sólo por su mayor simplicidad y por las opciones que PHP añade sino porque en caso de que se quisiera desarrollar en un futuro para otra plataforma la parte de la base de datos y su conexión con la aplicación se podría reutilizar.

#### *PHP*

PHP es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas. Se usa principalmente para la interpretación del lado del servidor.

#### **Características:**

- Orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una base de datos.
- Es considerado un lenguaje fácil de aprender, ya que en su desarrollo se simplificaron distintas especificaciones, como es el caso de la definición de las variables primitivas, ejemplo que se hace evidente en el uso de PHP arrays.
- El código fuente escrito en PHP es invisible al navegador web y al cliente ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador. Esto hace que la programación en PHP sea segura y confiable.
- Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y PostgreSQL.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- No requiere definición de tipos de variables aunque sus variables se pueden evaluar también por el tipo que estén manejando en tiempo de ejecución.

### 2.3.4 SERVIDOR

Aunque parezca una elección trivial, debido a la gran cantidad de servidores que hay disponibles, el servidor tendrá que tener por ahora dos características: que se pueda alojar una base de datos MySQL y que tenga PHP. Obviamente, como se ha descrito en las características que tenía que tener nuestra aplicación, también habrá que dar soporte tanto de imágenes como de mapas, pero para evitar cuellos de botella en el servidor y poder prestar un servicio rápido, estas características se proporcionarán mediante servicios web externos y gratuitos que se explicarán más adelante. Teniendo en cuenta esto, para el desarrollo de este



proyecto se utilizará una solución sencilla y cómoda, que permitirá un desarrollo más rápido. La solución elegida para ello será Mowes Portable.

### ***MOWES PORTABLE***

Mowes Portable es un software libre que no necesita instalación y que es de fácil manejo. Para descargarlo únicamente hay que navegar por su página seleccionando los elementos que se quieren incluir (en este caso servidor Apache, PHP y MySQL) y él solo crea un ZIP que al descomprimirlo se podrá utilizar como si fuera un programa que se hubiera instalado previamente.

### **2.3.5 SERVICIO WEB DE IMÁGENES**

Como se ha dicho anteriormente en este proyecto se ha decidido usar un servicio web para el alojamiento de las imágenes de las denuncias. Hay gran cantidad de servicios web para compartir imágenes pero teniendo en cuenta los que no incluyen publicidad, y observando el espacio de almacenamiento que proporcionan, se ha decidido reducir la lista a dos posibles candidatos: Flickr y Picasa. Este análisis se centrará principalmente en las dos características importantes para el proyecto: el acceso al servicio con cuentas de correo, la capacidad de almacenaje con cuentas gratuitas y la documentación de su API.

### ***FLICKR***

Flickr es un sitio web propiedad de Yahoo! que permite almacenar, ordenar, buscar, vender<sup>2</sup> y compartir fotografías y videos en línea.

- **Acceso al servicio:**

En cuanto a el acceso al servicio Flickr proporciona la posibilidad de utilizar una cuenta de Yahoo! con la cual se tendría acceso directamente, o utilizar una cuenta de Facebook o Gmail siempre y cuando se le den los permisos de acceso previamente.

- **Capacidad:**

En cuanto a la capacidad de almacenaje Flickr permite con una cuenta gratuita la posibilidad de almacenar 300 MB por cada mes del año. Además si se supera la cantidad de 200 fotografías Flickr únicamente mostrará en la galería las 200 últimas.

- **Documentación:**



La documentación de la API para desarrolladores de Flickr se encuentra disponible en un apartado de su página y está bien organizada mostrando un índice con las distintas operaciones que el desarrollador puede realizar con ella y según el lenguaje para el que se quiera desarrollar, el uso de esta API es gratuito para aplicaciones no comerciales aunque también se puede utilizar con fines comerciales previo acuerdo.

### **PICASA**

Picasa es un organizador y visor de imágenes, creado tanto para organizar como para editar fotografías digitales; además posee un sitio web integrado para compartir fotos. "Picasa" es una mezcla del apellido del pintor español Pablo Picasso, la frase en español de "mi casa" y "pic" de "pictures" o imágenes (la idea quiere decir "el arte personalizado"). Fue creada por Lifescape y en la actualidad es propiedad de Google.

- **Acceso al servicio:**

Para acceder a Picasa Web Albums es obligatorio tener una cuenta de Google, pero si ya se tiene no hace falta ningún tipo de registro ni activación de la cuenta, el usuario se puede identificar con su cuenta de Google y entrar directamente Picasa.

- **Capacidad:**

El almacenamiento de fotografías y videos en Picasa para cuentas gratuitas en Picasa es de 1 GB. Pero dentro de las cuentas gratuitas Picasa diferencia entre dos usuarios: los que tienen activado Google+ y los que no.

Para los primeros una vez se supera el almacenamiento máximo se podrán seguir subiendo fotos pero si su tamaño es mayor a 2.048 x 2.048 automáticamente se cambiará su tamaño a 2048 en su extremo más largo además las fotografías que sean inferiores a este tamaño no cuentan en su cuota. De esta manera se tiene una capacidad de almacenamiento ilimitada.

Para los segundos se cumplen las mismas características pero siendo el límite de resolución menor 800 x 800, pero igualmente se tiene una capacidad de almacenamiento ilimitada.

- **Documentación**

Al igual que Flickr, Picasa también provee al desarrollador de una API integrada en Google Developers la cual aunque esta peor organizada cumple su misma funcionalidad.

### **CONCLUSIÓN**

Como el proyecto se desarrollara para Android, esto implica que el usuario tenga una cuenta de Google para el acceso a Google Play, de donde se podrá descargar la aplicación



desarrollada en este proyecto. Teniendo en cuenta esto para que el usuario no tenga que activar ninguna cuenta más, ni registrarse en ningún otro lado que no sea la aplicación, Picasa proporciona un acceso más cómodo para el usuario.

Por otra parte, Picasa proporciona al usuario una capacidad de almacenamiento ilimitada, característica que Flickr no puede ofrecer.

Por todo ello para el alojamiento de fotografías en este proyecto se utilizará Picasa.

### 2.3.6 SERVIDOR DE MAPAS

Aunque existen varios servidores de mapas, como este proyecto se desarrollará para Android, el uso de Google Maps es prácticamente obligatorio debido a su gran integración con Android y la gran cantidad de documentación que proporciona Google de su API.

El único aspecto que podría llevar a no utilizar Google Maps sería que Google solo permite su utilización gratis para aplicaciones no comerciales, pero como no es el caso porque la aplicación a desarrollar será gratuita para llegar a la mayor cantidad de gente, no influirá en el proyecto.



## CAPÍTULO 3. ANÁLISIS Y DISEÑO DEL SISTEMA

### 3.1 CASOS DE USO

Este apartado realiza una primera aproximación a la funcionalidad deseada para el sistema mediante Casos de Uso. El modelado mediante casos de uso es una técnica utilizada para obtener los requisitos funcionales de un sistema, estructurados en torno a diferentes roles de usuarios participantes. Se basa en **UML** (*Unified Modeling Language*).

El siguiente esquema muestra la asociación entre los casos de uso del sistema y los actores que toman parte:

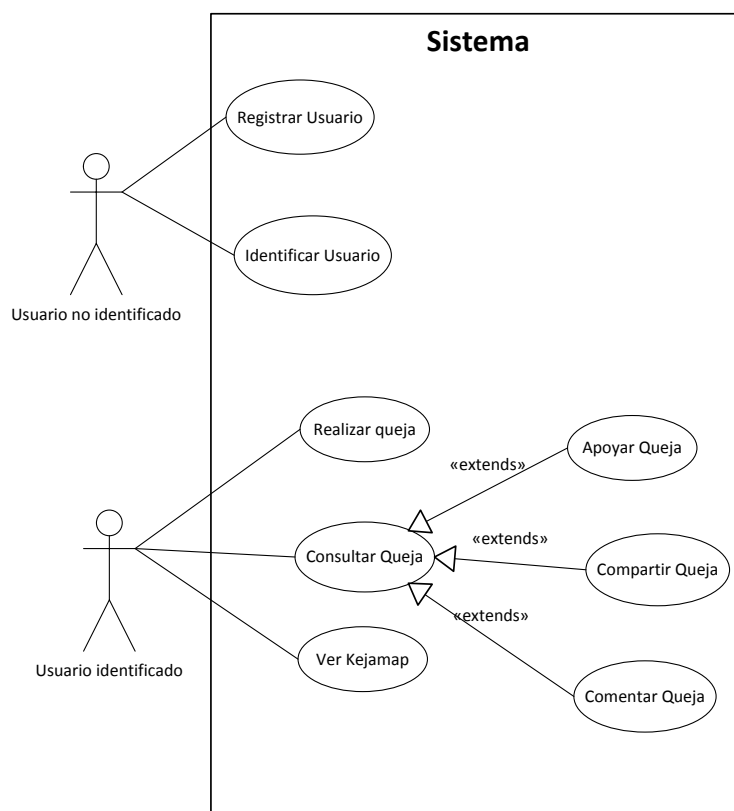


Tabla 4. Diagrama de Casos de Uso

A continuación se detalla la especificación de casos de uso. Cada Caso de Uso estará definido por los siguientes atributos:



- **Identificador:** Identifica al caso de uso de forma única. Debe seguir el formato: CU-XXX, siendo XXX un valor numérico único para cada caso de uso.
- **Nombre:** Breve especificación textual del caso de uso.
- **Actores:** Tipo de usuario del sistema que inicia el caso de uso.
- **Objetivo:** Finalidad del caso de uso.
- **Precondiciones:** Estado previo que se debe cumplir para poder realizar una operación.
- **Postcondiciones:** Estado en el que queda el sistema tras realizar una operación.
- **Escenario básico:** Especifica la manera en la que interactúa un actor con el sistema y cuál es la respuesta que el sistema le ofrece.
- **Escenarios alternativos:** Condiciones excepcionales que afectan al escenario y respuestas del sistema ante esas situaciones.

Identificador	CU-001
Nombre	Registrar usuario
Actores	Usuario no identificado
Descripción	El usuario no registrado se da de alta en el sistema.
Precondiciones	Nombre de usuario no utilizado, veracidad en los datos de registro.
Postcondiciones	Usuario registrado en el sistema con los permisos correspondientes.
Escenario Básico	<ol style="list-style-type: none"><li>1. El usuario no registrado escoge la opción “Registrarse”.</li><li>2. Introduce la información básica necesaria, incluyendo nombre de usuario, correo de gmail y contraseña del correo, que será la misma que la del inicio de sesión.</li><li>3. El sistema comprueba la validez de los datos introducidos.</li><li>4. El sistema almacena la información del usuario.</li><li>5. El sistema comunica al usuario no registrado el éxito en el registro.</li></ol>
Escenario Altern.	<ol style="list-style-type: none"><li>3b1. El sistema comunica al usuario no registrado que los datos no son válidos.</li><li>3b2. Vuelta al paso2 para volver a introducirlos datos correctamente.</li></ol>



Tabla 5. Caso de Uso 001: Registro de usuario

Identificador CU-002	
Nombre	Identificar usuario
Actores	Usuario no identificado
Descripción	El usuario se identifica para acceder al sistema.
Precondiciones	El usuario debe estar registrado en el sistema.
Postcondiciones	El usuario pasa a ser un usuario identificado.
Escenario Básico	<ol style="list-style-type: none"><li>1. El usuario introduce el nombre de usuario y contraseña y selecciona “Acceder”.</li><li>2. El sistema comprueba los datos introducidos.</li><li>3. El sistema muestra un menú para que pueda elegir las diferentes opciones.</li></ol>
Escenario Altern.	<p>3b1. El sistema comunica al usuario que los datos introducidos no son correctos.</p> <p>3b2. Vuelta al paso 1 para introducirlos datos de nuevo.</p>

Tabla 6. Caso de Uso 002: Identificar usuario

Identificador CU-003	
Nombre	Realizar queja
Actores	Usuario identificado
Descripción	El usuario registra una denuncia en el sistema.
Precondiciones	Haber seleccionado la opción de Realizar queja
Postcondiciones	Denuncia registrada.
Escenario Básico	<ol style="list-style-type: none"><li>1. El usuario elige la opción de realizar queja.</li><li>2. El usuario completa los datos de la queja.</li><li>3. El sistema registra la queja.</li><li>4. El sistema avisa al ciudadano que la denuncia ha sido realizada.</li></ol>
Escenario Altern.	<p>3b1. El sistema comunica al ciudadano en caso de error en el registro de denuncia.</p> <p>3b2. Vuelta al paso 2 para registrar la denuncia nuevamente.</p>

Tabla 7. Caso de Uso 003: Realizar queja



Identificador	CU-004
Nombre	Consultar Queja
Actores	Usuario identificado.
Descripción	El usuario realiza una consulta a partir de una serie de campos sobre el sistema.
Precondiciones	Ninguna.
Postcondiciones	El sistema recibe la consulta y realiza las operaciones necesarias.
Escenario Básico	<ol style="list-style-type: none"><li>1. El usuario elige la opción de “Consultar quejas”.</li><li>2. El ciudadano rellena el formulario con los campos por los que desee hacer la consulta.</li><li>3. El sistema devuelve las quejas conformes a la consulta realizada por el usuario.</li></ol>
Escenario Altern.	<p>2b1. El sistema no encuentra ninguna queja con los parámetros dados.</p> <p>2b2. El sistema comunica al usuario que no encuentra ninguna queja.</p>

Tabla 8. Caso de Uso 004: Consultar queja

Identificador	CU-005
Nombre	Ver kejamap
Actores	Usuario identificado.
Descripción	El usuario visualiza un mapa con las quejas geoposicionadas.
Precondiciones	Ninguna.
Postcondiciones	El sistema recibe la consulta y realiza las operaciones necesarias.
Escenario Básico	<ol style="list-style-type: none"><li>1. El usuario elige la opción de “Ver Kejamap”.</li><li>2. El sistema carga el mapa con las quejas.</li><li>3. El usuario navega a través del mapa.</li></ol>
Escenario Altern.	

Tabla 9. Caso de Uso 005: Ver Kejamap

Identificador	CU-006
Nombre	Apoyar queja
Actores	Usuario identificado.



<b>Descripción</b>	Permite al usuario apoyar una queja.
<b>Precondiciones</b>	Haber seleccionado una queja
<b>Postcondiciones</b>	El sistema registra el apoyo asociado a la queja, incrementa el número de apoyos y deshabilita la opción de apoyar queja.
<b>Escenario Básico</b>	<ol style="list-style-type: none"><li>1. El usuario selecciona una queja.</li><li>2. El usuario selecciona “apoyar queja”.</li><li>3. El sistema registra el apoyo de la queja.</li><li>4. El sistema incrementa el contador de apoyos de la queja.</li><li>5. El sistema deshabilita la opción de apoyar queja.</li></ol>
<b>Escenario Altern.</b>	

Tabla 10. Caso de Uso 006: Apoyar queja

<b>Identificador</b>	<b>CU-007</b>
<b>Nombre</b>	Comentar queja
<b>Actores</b>	Usuario identificado.
<b>Descripción</b>	Permite al usuario comentar una queja seleccionada
<b>Precondiciones</b>	Haber seleccionado una queja
<b>Postcondiciones</b>	El sistema registra el comentario asociado a la queja.
<b>Escenario Básico</b>	<ol style="list-style-type: none"><li>1. El usuario selecciona una queja.</li><li>2. El usuario selecciona “comentar queja” y rellena el cuadro de texto.</li><li>3. El sistema registra el comentario de la queja.</li></ol>
<b>Escenario Altern.</b>	<ol style="list-style-type: none"><li>2b. El usuario no introduce nada en el cuadro de texto.</li><li>3.b El sistema comunica al usuario que introduzca algo.</li></ol>

Tabla 11. Caso de Uso 007: Comentar queja

<b>Identificador</b>	<b>CU-008</b>
<b>Nombre</b>	Compartir queja
<b>Actores</b>	Usuario identificado.
<b>Descripción</b>	Permite al usuario compartir una queja.
<b>Precondiciones</b>	Haber seleccionado una queja
<b>Postcondiciones</b>	El sistema registra el comentario asociado a la queja.
<b>Escenario Básico</b>	<ol style="list-style-type: none"><li>1. El usuario selecciona una queja.</li><li>2. El usuario selecciona “compartir queja”.</li><li>3. El sistema muestra al usuario las distintas opciones de compartir la queja.</li><li>4. El usuario selecciona una de las opciones.</li></ol>



5. El sistema comparte la queja.

Escenario Altern.

Tabla 12. Caso de Uso 008: Compartir queja

## 3.2 ESPECIFICACIÓN DE REQUISITOS

Este apartado cubre la especificación de requisitos del sistema. Los requisitos se han dividido en dos grupos: requisitos funcionales y requisitos no funcionales. Para estos últimos, además, se realizan una subdivisión por categorías.

Para la presentación de los requisitos se ha utilizado la plantilla *Volere*. Los campos que incluye la plantilla son:

- **ID Requisito:** Identificador unívoco del requisito.
- **Tipo:** Tipo de requisito (funcional, de usabilidad, de rendimiento...)
- **Evento / Casos de uso:** Eventos o casos de uso relacionados con el requisito
- **Descripción:** Objetivo del requisito.
- **Justificación:** La razón de por qué es necesario incluir el requisito.
- **Origen:** Persona que ha planteado la necesidad del requisito
- **Criterio de cumplimiento:** La manera de medir si el requisito se ha cubierto en la implementación de la aplicación.
- **Satisfacción del cliente:** Valor entre 1 (indiferente) y 5 (muy satisfecho) que indique el grado de satisfacción que le produce al cliente que se cumpla el requisito.
- **Insatisfacción del cliente:** Valor entre 1 (indiferente) y 5 (muy satisfecho) que indique el grado de insatisfacción que le produce al cliente que no se cumpla el requisito.
- **Prioridad:** Prioridad del requisito según el cliente.
- **Conflictos:** Conflictos que se produce con otros requisitos si se implementa el requisito.
- **Material de soporte:** Documentación adicional que explique el requisito.
- **Historia:** Cuándo se crea el requisito y los cambios que se han realizado sobre éste.



REQUISITOS FUNCIONALES:

ID Requisito	RFN-001	Tipo:	Funcional
Evento / Caso de Uso	CU-001		
Descripción	Registro de Usuarios		
Justificación	Es necesario almacenar la información relativa a los usuarios para su posterior identificación.		
Origen	Desarrollador / Cliente		
Criterio de cumplimiento	Comprobación de la información del usuario registrado mantenida en el sistema y la posibilidad de modificación y borrado		
Satisfacción del cliente	4	Insatisfacción del cliente	4
Prioridad	ALTA	Conflictos	NINGUNO
Material de soporte			
Historia	Creado el 14.3.2011		

Tabla 13. RFN-001: Registro de Usuarios

ID Requisito	RFN-002	Tipo:	Funcional
Evento / Caso de Uso	CU-002		
Descripción	Identificación de Usuarios		
Justificación	Para acceder a las funcionalidades que proporciona el usuario tendrá que identificarse.		
Origen	Desarrollador / Cliente		
Criterio de cumplimiento	Que un usuario ya registrado sea capaz de acceder al sistema introduciendo sus datos de identificación.		
Satisfacción del cliente	4	Insatisfacción del cliente	4
Prioridad	ALTA	Conflictos	NINGUNO
Material de soporte			
Historia	Creado el 14.3.2011		

Tabla 14. RFN-002: Identificación de Usuarios

ID Requisito	RFN-003	Tipo:	Funcional
Evento / Caso de Uso	CU-003		
Descripción	Registro de queja		
Justificación	El usuario debe tener la posibilidad de realizar una queja y que ésta quede registrada en el sistema.		
Origen	Desarrollador / Cliente		



Criterio de cumplimiento	Que un usuario realice una queja y que posteriormente se compruebe que esta aparece en la BBDD.		
Satisfacción del cliente	4	Insatisfacción del cliente	4
Prioridad	ALTA	Conflictos	NINGUNO
Material de soporte			
Historia	Creado el 14.3.2011		

Tabla 15. RFN-003: Registro de queja

ID Requisito	RFN-004	Tipo:	Funcional
Evento / Caso de Uso	CU-003		
Descripción	Selección de imagen		
Justificación	El usuario debe tener la posibilidad de adjuntar una imagen de su móvil a la queja.		
Origen	Desarrollador / Cliente		
Criterio de cumplimiento	Que el usuario realice la acción y comprobar que la imagen queda registrada en la base de datos.		
Satisfacción del cliente	4	Insatisfacción del cliente	4
Prioridad	ALTA	Conflictos	NINGUNO
Material de soporte			
Historia	Creado el 14.3.2011		

Tabla 16. RFN-004: Selección de imagen

ID Requisito	RFN-005	Tipo:	Funcional
Evento / Caso de Uso	CU-003		
Descripción	Selección de dirección.		
Justificación	El usuario debe tener la posibilidad de seleccionar la dirección de la queja en un mapa.		
Origen	Desarrollador / Cliente		
Criterio de cumplimiento	Que el usuario realice la acción y que una vez realizada la queja la posición de la queja se corresponda con la seleccionada.		
Satisfacción del cliente	4	Insatisfacción del cliente	4
Prioridad	ALTA	Conflictos	NINGUNO
Material de soporte			
Historia	Creado el 14.3.2011		

Tabla 17. RFN-005: Selección de dirección





ID Requisito	RFN-006	Tipo:	Funcional
Evento / Caso de Uso	CU-004		
Descripción	Consulta de quejas		
Justificación	El usuario debe tener la posibilidad de consultar quejas según los campos seleccionados en el formulario		
Origen	Desarrollador / Cliente		
Criterio de cumplimiento	Que un usuario realice una consulta y que la respuesta de la base de datos se corresponda con ella.		
Satisfacción del cliente	4	Insatisfacción del cliente	4
Prioridad	ALTA	Conflictos	NINGUNO
Material de soporte			
Historia	Creado el 14.3.2011		

Tabla 18. RFN-006: Consulta de quejas

ID Requisito	RFN-007	Tipo:	Funcional
Evento / Caso de Uso	CU-004		
Descripción	Vista de la lista de quejas		
Justificación	El usuario debe tener la posibilidad de observar una lista con las quejas correspondientes a su consulta anterior		
Origen	Desarrollador / Cliente		
Criterio de cumplimiento	Que un usuario realice una consulta y que la lista mostrada sea la correspondiente a su consulta		
Satisfacción del cliente	4	Insatisfacción del cliente	4
Prioridad	ALTA	Conflictos	NINGUNO
Material de soporte			
Historia	Creado el 14.3.2011		

Tabla 19. RFN-007: Vista de la lista de quejas

ID Requisito	RFN-008	Tipo:	Funcional
Evento / Caso de Uso	CU-004		
Descripción	Vista de queja seleccionada		
Justificación	El usuario debe tener la posibilidad de observar todos los datos de una queja seleccionada anteriormente		
Origen	Desarrollador / Cliente		
Criterio de cumplimiento	Comprobar que los datos mostrados de la queja se corresponden con la queja que se ha seleccionado anteriormente.		



## SISTEMA COLABORATIVO PARA LA DIVULGACIÓN DE INCIDENCIAS CIUDADANAS BASADO EN ANDROID

Satisfacción del cliente	4	Insatisfacción del cliente	4
Prioridad	ALTA	Conflictos	NINGUNO
Material de soporte			
Historia	Creado el 14.3.2011		

Tabla 20. RFN-008: Vista de queja seleccionada

ID Requisito	RFN-009	Tipo:	Funcional
Evento / Caso de Uso	CU-006		
Descripción	Apoyo de queja		
Justificación	El usuario debe tener la posibilidad de apoyar una vez una queja seleccionando un botón.		
Origen	Desarrollador / Cliente		
Criterio de cumplimiento	Que el usuario apoye una queja y comprobar que esta quede registrada en la base de datos.		
Satisfacción del cliente	4	Insatisfacción del cliente	4
Prioridad	ALTA	Conflictos	NINGUNO
Material de soporte			
Historia	Creado el 14.3.2011		

Tabla 21. RFN-009: Apoyo de queja

ID Requisito	RFN-010	Tipo:	Funcional
Evento / Caso de Uso	CU-07		
Descripción	Comentario de queja		
Justificación	El usuario debe tener la posibilidad de realizar comentarios sobre la queja seleccionada.		
Origen	Desarrollador / Cliente		
Criterio de cumplimiento	Que el usuario realice un comentario sobre una queja y comprobar que este se registra en la base de datos.		
Satisfacción del cliente	4	Insatisfacción del cliente	4
Prioridad	ALTA	Conflictos	NINGUNO
Material de soporte			
Historia	Creado el 14.3.2011		

Tabla 22. RFN-010: Comentario de queja



*REQUISITOS NO FUNCIONALES:*

<b>ID Requisito</b>	RUS-001	<b>Tipo:</b>	Usabilidad
<b>Evento / Caso de Uso</b>	Todos los CU		
<b>Descripción</b>	El vocabulario utilizado por la aplicación será entendible por personas con nivel cultural bajo.		
<b>Justificación</b>	El vocabulario será sencillo y se evitarán los tecnicismos y cultismos para que sea entendible por los usuarios.		
<b>Origen</b>	Cliente		
<b>Criterio de cumplimiento</b>	Solicitar a usuarios con un nivel cultural bajo que hagan uso de la aplicación y cerciorarse de que, mínimo el 85% de los usuarios, entiendan perfectamente el vocabulario de la aplicación.		
<b>Satisfacción del cliente</b>	5	<b>Insatisfacción del cliente</b>	4
<b>Prioridad</b>	ALTA	<b>Conflictos</b>	NINGUNO
<b>Material de soporte</b>			
<b>Historia</b>	Creado el 13.3.2011		

Tabla 23. RUS-001

<b>ID Requisito</b>	RUS-002	<b>Tipo:</b>	Usabilidad
<b>Evento / Caso de Uso</b>	Todos los CU		
<b>Descripción</b>	Los ciudadanos se familiarizarán rápidamente con la aplicación.		
<b>Justificación</b>	Los ciudadanos podrán familiarizarse con la aplicación en un corto periodo de tiempo, incluso las personas con pocos conocimientos informáticos.		
<b>Origen</b>	Cliente		
<b>Criterio de cumplimiento</b>	Observar que un ciudadano con pocos conocimientos informáticos puede aprender a utilizar el 75% de la funcionalidad que ofrece el sistema en menos de 30 minutos		
<b>Satisfacción del cliente</b>	4	<b>Insatisfacción del cliente</b>	4
<b>Prioridad</b>	ALTA	<b>Conflictos</b>	NINGUNO
<b>Material de soporte</b>			
<b>Historia</b>	Creado el 13.3.2011		

Tabla 24. RUS-002



ID Requisito	RUS-003	Tipo:	Usabilidad
Evento / Caso de Uso	Todos los CU		
Descripción	Funcionamiento fácilmente recordable.		
Justificación	El funcionamiento del sistema será fácilmente recordable por los usuarios debido a que la aplicación es de uso ocasional y no habitual, y los usuarios pueden estar bastante tiempo sin utilizarla.		
Origen	Cliente		
Criterio de cumplimiento	Observar que un usuario recuerda cómo utilizar el sistema después de más de un mes si acceder a éste.		
Satisfacción del cliente	4	Insatisfacción del cliente	3
Prioridad	ALTA	Conflictos	NINGUNO
Material de soporte			
Historia	Creado el 13.3.2011		

Tabla 25. RUS-003

ID Requisito	RUS-004	Tipo:	Usabilidad
Evento / Caso de Uso	Todos los CU.		
Descripción	La aplicación estará orientada a que el usuario no cometa errores		
Justificación	Sí al usuario se le aborda con mensajes de error y correcciones puede influir en su satisfacción de la aplicación.		
Origen	Cliente		
Criterio de cumplimiento	La aplicación muestra avisos al usuario cuando introduce datos erróneos o realice acciones no permitidas. Se debe comprobar que al dejarle utilizar a la aplicación a un conjunto de usuarios, les avise de al menos el 95% de los errores que comete.		
Satisfacción del cliente	5	Insatisfacción del cliente	4
Prioridad	ALTA	Conflictos	NINGUNO
Material de soporte			
Historia	Creado el 13.3.2011		

Tabla 26. RUS-004

ID Requisito	RUS-005	Tipo:	Usabilidad
Evento / Caso de Uso	CU-006, CU-007, CU-008		
Descripción	El sistema fomentará la repetida utilización de éste por parte de los usuarios mediante el reconocimiento de su colaboración.		
Justificación	Un usuario al que se le reconozca su colaboración, estará más satisfecho con la aplicación y creará un vínculo con ella mayor.		
Origen	Cliente		



<b>Criterio de cumplimiento</b>	Se realizará una estadística del número de veces que acceden los usuarios a la aplicación, y se comprobará que al menos el 90% vuelva a utilizarla.		
<b>Satisfacción del cliente</b>	5	<b>Insatisfacción del cliente</b>	5
<b>Prioridad</b>	ALTA	<b>Conflictos</b>	NINGUNO
<b>Material de soporte</b>			
<b>Historia</b>	Creado el 13.3.2011		

Tabla 27. RUS-005

<b>ID Requisito</b>	RUS-006	<b>Tipo:</b>	Usabilidad
<b>Evento / Caso de Uso</b>	Todos los CU		
<b>Descripción</b>	Para cualquier operación. Se le mostrará un mensaje al usuario indicando si se ha realizado correctamente o no.		
<b>Justificación</b>	Impedir que el usuario piense que la aplicación no responde a su interacción.		
<b>Origen</b>	Cliente		
<b>Criterio de cumplimiento</b>	Comprobar que en todas las operaciones elegidas se muestra el mensaje al usuario.		
<b>Satisfacción del cliente</b>	5	<b>Insatisfacción del cliente</b>	5
<b>Prioridad</b>	ALTA	<b>Conflictos</b>	NINGUNO
<b>Material de soporte</b>			
<b>Historia</b>	Creado el 12.3.2011		

Tabla 28. RUS-006

<b>ID Requisito</b>	RAP-001	<b>Tipo:</b>	Apariencia
<b>Evento / Caso de Uso</b>	Todos los CU		
<b>Descripción</b>	La aplicación deberá guardar una línea de diseño en cuanto a botones y pantallas parecida a la del sistema operativo para el que se implemente.		
<b>Justificación</b>	Para que al usuario sea más fácil familiarizarse con la aplicación.		
<b>Origen</b>	Sistema operativo		
<b>Criterio de cumplimiento</b>	Comprobar que no se ha utilizado elementos de la interfaz no incluidos en la API del sistema operativo		
<b>Satisfacción del cliente</b>	5	<b>Insatisfacción del cliente</b>	5
<b>Prioridad</b>	ALTA	<b>Conflictos</b>	NINGUNO
<b>Material de soporte</b>	Principios de diseño del sistema operativo		
<b>Historia</b>	Creado el 12.3.2011		

Tabla 29. RAP-001



ID Requisito	RRE-001	Tipo:	Rendimiento
Evento / Caso de Uso	CU-004		
Descripción	En el listado de quejas no mostrar más de 5 quejas a la vez.		
Justificación	Al contener miniaturas de las fotografías se podría saturar el móvil y que la aplicación dejara de funcionar.		
Origen	Desarrollador		
Criterio de cumplimiento	Comprobar que no se muestran más quejas del el límite impuesto		
Satisfacción del cliente	5	Insatisfacción del cliente	5
Prioridad	ALTA	Conflictos	NINGUNO
Material de soporte			
Historia	Creado el 12.3.2011		

Tabla 30. RRE-001

ID Requisito	RSG-001	Tipo:	Seguridad
Evento / Caso de Uso	Todos los CU		
Descripción	El acceso a la base de datos se realizará mediante un usuario y una clave transparentes al usuario.		
Justificación	Evitar el acceso de personas ajenas a la aplicación a la base de datos.		
Origen	Desarrollador		
Criterio de cumplimiento	Comprobar que no se pueda acceder a las funciones de la aplicación sin haber realizado la operación de identificación.		
Satisfacción del cliente	4	Insatisfacción del cliente	4
Prioridad	ALTA	Conflictos	NINGUNO
Material de soporte			
Historia	Creado el 19.3.2011		

Tabla 31. RSG-001

ID Requisito	RLE-001	Tipo:	Legales
Evento / Caso de Uso	Todos los CU		
Descripción	El único dato personal del usuario que se almacenará en la base de datos será el correo de Gmail.		
Justificación	No tener problemas con el incumplimiento de la LOPD.		

<b>Origen</b>	Cliente		
<b>Criterio de cumplimiento</b>	Comprobar que se evita que el usuario proporcione más datos personales.		
<b>Satisfacción del cliente</b>	5	<b>Insatisfacción del cliente</b>	5
<b>Prioridad</b>	ALTA	<b>Conflictos</b>	NINGUNO
<b>Material de soporte</b>			
<b>Historia</b>	Creado el 12.3.2011		

Tabla 32. RLE-001

### 3.3 DIAGRAMA DE CLASES

A continuación se pasará a mostrar los diagramas de clases, tanto de la base de datos como de la aplicación Android. Se han decidido no añadir los métodos de ninguna clase debido a que son muy numerosos y no aportan gran significado a la naturaleza de las clases y a sus relaciones, aun así más adelante en el capítulo 4 se explicarán los más importantes.

#### 3.3.1 DIAGRAMA DE CLASES DE LA BASE DE DATOS

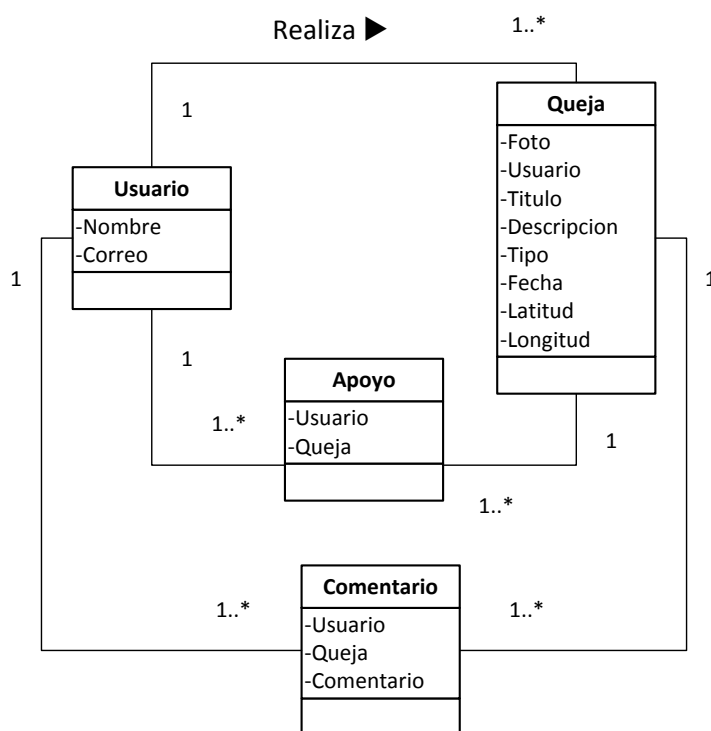


Ilustración 14. Diagrama de clases de la base de datos

### 3.3.2 DIAGRAMA DE CLASES DE LA APLICACIÓN ANDROID

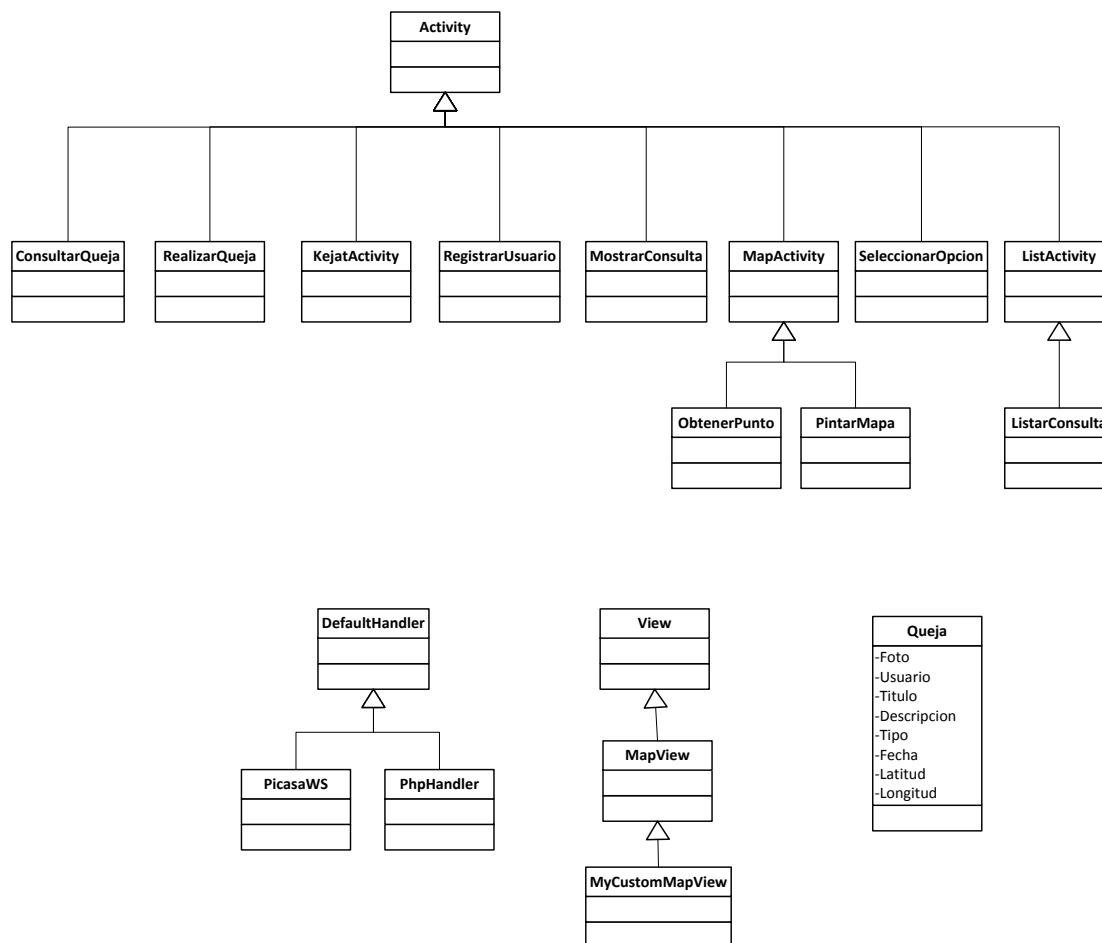


Ilustración 15. Diagrama de clases de la aplicación Android



### 3.4 DIAGRAMAS DE SECUENCIA

En este apartado se mostrarán los diagramas de secuencia de los distintos casos de uso, estos darán una idea de cómo funciona la aplicación aunque se explicará más detalladamente en el capítulo cuatro.

#### 3.4.1 DIAGRAMA DE SECUENCIA DE REGISTRAR USUARIO

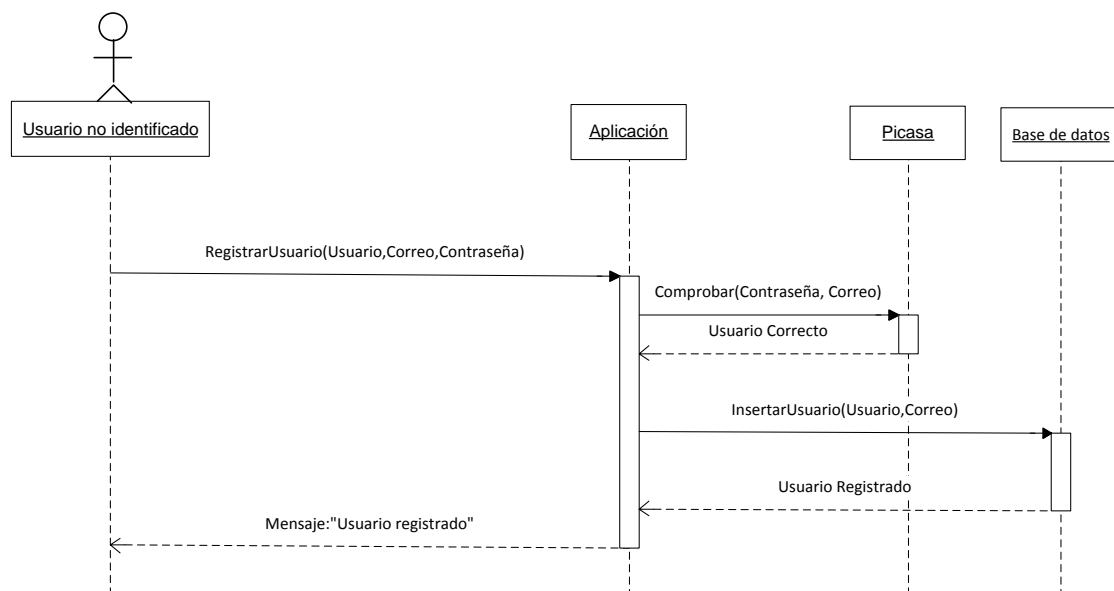


Ilustración 16. Diagrama de secuencia de registrar usuario

### 3.4.2 DIAGRAMA DE SECUENCIA DE IDENTIFICAR USUARIO

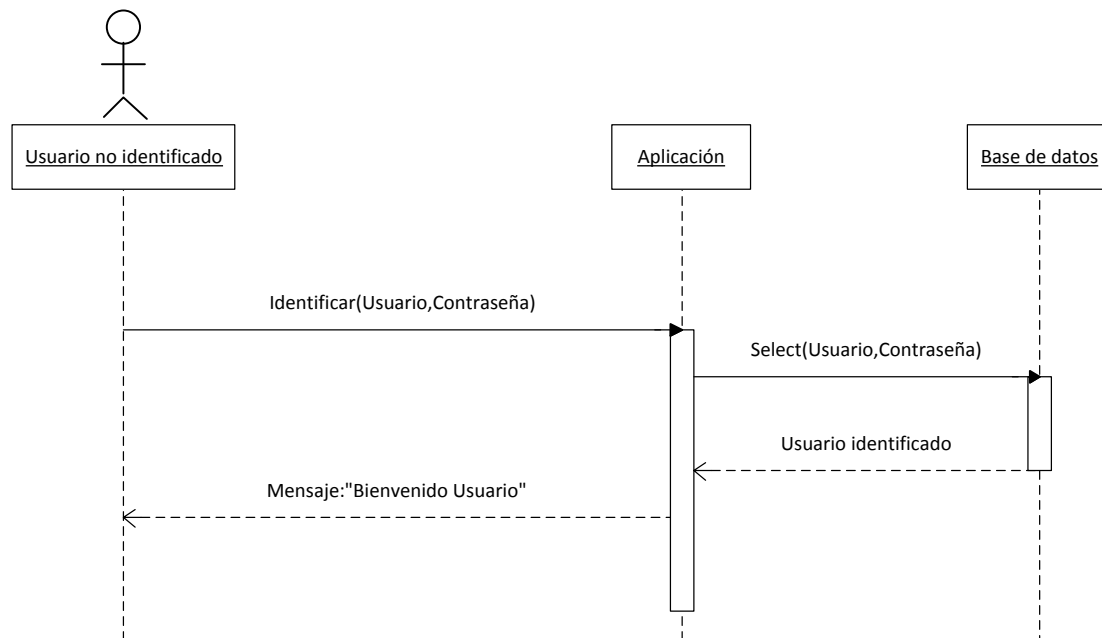


Ilustración 17. Diagrama de secuencia de identificar usuario

### 3.4.3 DIAGRAMA DE SECUENCIA DE VER KEJAMAP

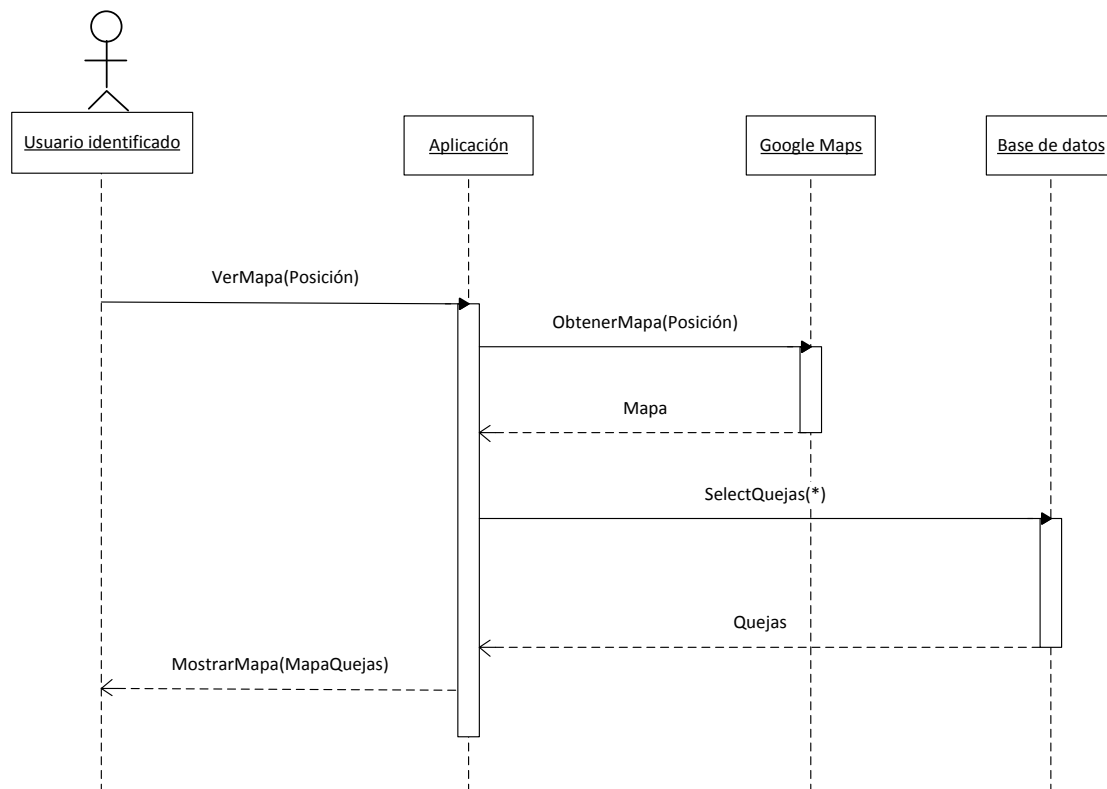


Ilustración 18. Diagrama de secuencia de verkejamap

### 3.4.4 DIAGRAMA DE SECUENCIA DE REALIZARQUEJA

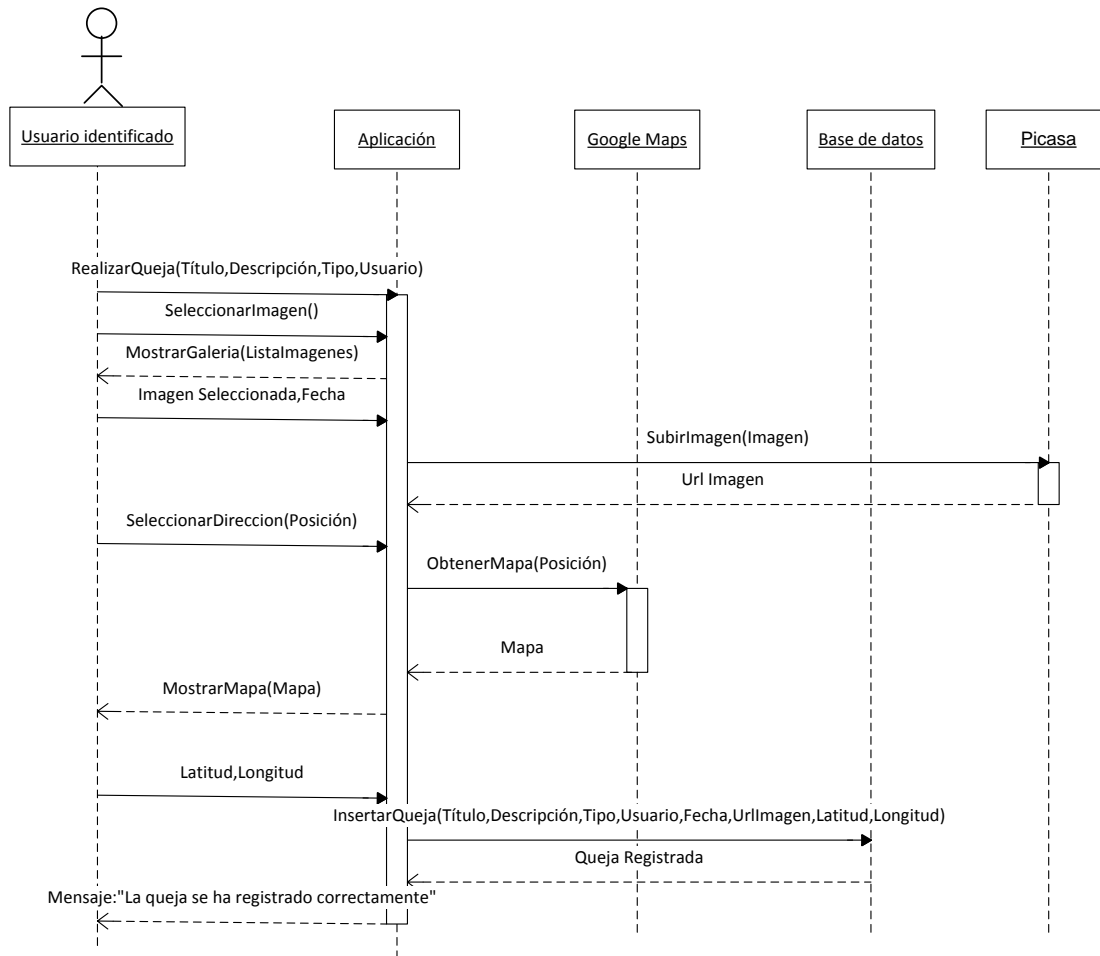


Ilustración 19. Diagrama de secuencia de realizarqueja

### 3.4.5 DIAGRAMA DE SECUENCIA DE CONSULTAR QUEJAS, APOYAR QUEJA Y REALIZAR COMENTARIO

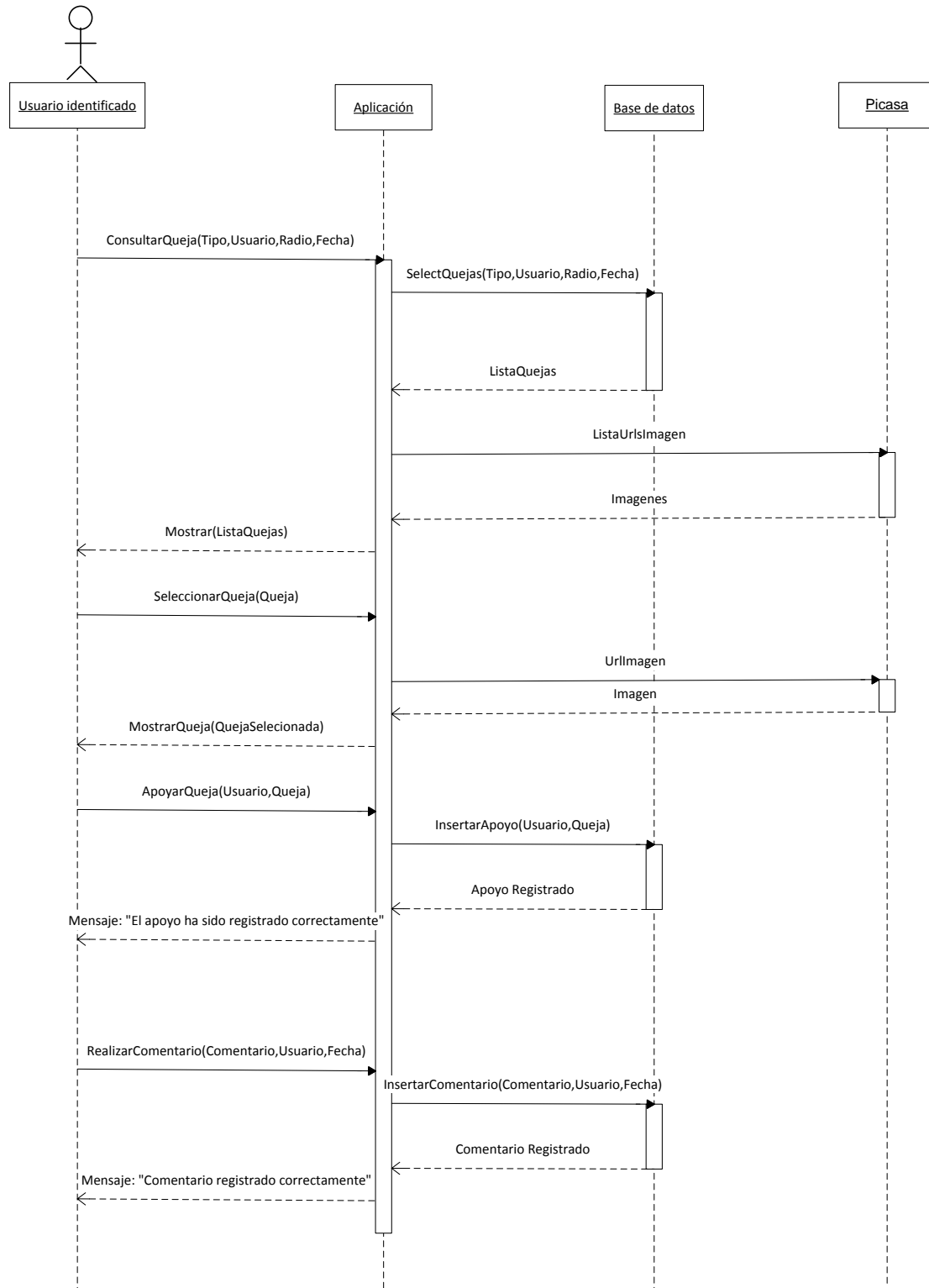


Ilustración 20. Diagrama de secuencia de consultarqueja, comentarqueja y apoyarqueja



## CAPÍTULO 4. IMPLEMENTACIÓN

### 4.1 SERVIDOR

En este apartado se mostrarán las distintas partes de la implementación del lado del servidor. Como se ha dicho anteriormente para ello se utilizara Mowes Portable que es una solución compuesta por un servidor Apache, un sistema de gestión de BBDD MySQL y PHP 5.0.

#### 4.1.1 SISTEMA DE GESTIÓN DE BASE DE DATOS

##### *INTRODUCCIÓN*

En este apartado se describirá como se ha implementado la aplicación, primero se comenzará describiendo como se ha implementado la base de datos para poder tener una idea de cómo se ha establecido la estructura de los datos, y después se continuará con las distintas partes de la implementación de Android, junto con los PHP que utilizan y que les permiten acceder a los datos almacenados en la BBDD.

La base de datos se desarrollará en MySQL y aunque se ha visto en el estudio del arte que la solución más rápida y cómoda es la utilización de Mowes Portable, sería trivial portar la base de datos a cualquier otro servidor apache que tenga PHP.

La decisión de elegir MySQL, es debido a que al tratarse de una base de datos pequeña (únicamente 4 tablas), no se necesita para mantenerla las distintas opciones y gran cantidad de configuraciones que permiten otras bases de datos, como por ejemplo Oracle .

## MYSQL

A continuación se pasará a mostrar la base de datos en detalle mostrando las tablas con cada uno de sus atributos e indicando su naturaleza.

<b>Nombre de tabla:</b>	Usuarios
<b>Descripción:</b>	Contiene la información relativa a los usuarios

Tabla 33. BBDD Tabla Usuarios

Field	Type	Null	Key	Default	Extra
correo	char(50)	NO	PRI	NULL	
nombre	char(50)	NO		NULL	

Ilustración 21. BBDD Tabla Usuarios

En esta tabla se almacena el correo (gmail) del usuario y el nombre con el que se registra, de esta manera se evita el asociar directamente una cuenta a una queja y que salga el correo del usuario al mostrar quién ha realizado una determinada queja. Además se ha decidido no almacenar la contraseña del correo del usuario para evitar problemas legales con la utilización de datos y para que la aplicación no sea objetivo de ataques maliciosos, de esta manera el usuario se sentirá mas cómodo sin tener que dar ningún dato personal que pueda comprometerle o que pueda ser utilizado por segundas personas.

Los dos campos son del tipo de cadena de caracteres y tienen como longitud máxima 50 caracteres. Aunque se ha optado porque la clave primaria sea el correo, la elección es trivial debido a que cualquiera de las dos podría ser, debido a que las dos son únicas por cada usuario. Además aunque pudiera parecer una inconsistencia la no nulidad de los campos y que el valor por defecto sea "NULL" nunca se va a dar este caso debido a que siempre en las inserciones se contralará esto mediante un archivo PHP.

<b>Nombre de tabla:</b>	Quejas
<b>Descripción:</b>	Contiene la información relativa a cada una de las quejas.

Tabla 34. BBDD Tabla Quejas

Field	Type	Null	Key	Default	Extra
usuario	varchar(50)	YES		NULL	
título	varchar(100)	YES		NULL	
foto	char(200)	NO	PRI	NULL	
descripcion	varchar(160)	YES		NULL	
tipo	varchar(50)	YES		NULL	
fecha	date	YES		NULL	
latitud	float(10,6)	YES		NULL	
ciudad	varchar(50)	YES		NULL	
longitud	float(10,6)	YES		NULL	

Ilustración 22. BBDD Tabla Quejas

En esta tabla se almacena todos los datos relativos a cada denuncia, los cuales se describen a continuación:

- El campo “usuario” es el nombre del usuario que ha realizado la queja, por ello tiene la misma longitud que el campo nombre en la tabla anterior.
- El título y la descripción contienen una cadena de caracteres que permite añadir información a la foto asociada.
- El campo “foto” contiene la *url* de Picasa donde está alojada la foto, también se utiliza de clave primaria debido a que es el único campo que es distinto en todas las quejas.
- El campo “tipo” indica el tipo de denuncia y puede adoptar unos valores ya definidos, (“vandalismo”, “deterioro de instalaciones públicas”, “desperfecto en la calzada” u “otro”). Siendo así se podría tomar la opción de utilizar un tipo enumerado en vez de una cadena de caracteres, pero si en un futuro se quieren añadir o cambiar esos valores predefinidos resulta menos complejo utilizar cadenas de caracteres.
- La “fecha” contiene la fecha en que se ha realizado la foto. Se ha utilizado un tipo “date” en vez de un tipo numérico para que sea más eficiente a la hora de realizar consultas sobre ellas.
- Tanto para la “latitud” como para la “longitud” se ha elegido un tipo “float” con 6 decimales para no tener que truncar o redondear la posición obtenida del geotiquetado de las fotografías y que su representación en el mapa sea más exacta.
- Por último el campo “ciudad” en este caso no se utiliza actualmente pero se ha decidido incluir, para que sea más fácil la realización de trabajos futuros los cuales se explicarán en este documento más adelante.



<b>Nombre de tabla:</b>	Apoyos
<b>Descripción:</b>	Contiene la información sobre los apoyos a las denuncias realizados por los usuarios

Tabla 35. BBDD Tabla Apoyos

Field	Type	Null	Key	Default	Extra
usuario	varchar(50)	NO	PRI		
queja	varchar(200)	NO	PRI		

Ilustración 23. BBDD Tabla Apoyos

El campo “usuario” indica el nombre del usuario que ha realizado el apoyo y el campo “queja” contiene la clave identificativa de la queja, que como se ha indicado en la tabla anterior es la *url* de la imagen de la queja.

Al tratarse de una de una relación uno a uno entre las clases usuario y queja y no tener ningún atributo más, se podría haber optado por incluir un campo más en la tabla de usuarios y eliminar esta tabla, pero no se ha optado por esta opción debido a que se sobrecargaría la tabla de usuarios y esto repercutiría tanto en las consultas de los apoyos como en las de los usuarios.

<b>Nombre de tabla:</b>	Comentarios
<b>Descripción:</b>	Contiene los comentarios de las quejas realizados por los usuarios

Tabla 36. BBDD Tabla Comentarios

Field	Type	Null	Key	Default	Extra
fecha	timestamp	NO	PRI	CURRENT_TIMESTAMP	on update CURRENT
queja	varchar(200)	NO	PRI		
usuario	varchar(50)	NO	PRI		
comentario	varchar(160)	YES		NULL	

Ilustración 24. BBDD Tabla Comentarios

En esta tabla también se representa una relación entre el usuario y la queja pero a través de un comentario.

Tanto el campo “queja” como el campo “usuario” son las claves identificativas de sus tablas, el campo “comentario” contiene el texto introducido por el usuario en la aplicación cuando comenta una queja, y la fecha es de tipo “timestamp”, es decir, contiene el día y la hora de cuando se ha creado el comentario por el usuario. Esta fecha es rellenada automáticamente por la base de datos cuando insertan los demás campos.

## 4.1.2 PHP

En esta parte se describirá el funcionamiento de un archivo PHP y cual son sus funciones en este proyecto.

### *FUNCIONAMIENTO DE UN PHP.*

PHP es un lenguaje que permite gran cantidad de libertad a la hora de realizar operaciones [2]. En este proyecto se describirán sus funciones para acceder a una base de datos. Para ello se desarrollará uno de los utilizados, en este caso será “consultageneral.php” por su simplicidad.

```
1 <?php
2 $con = mysql_connect("localhost","android","pass_android");
3 if (!$con)
4 {
5     die('Could not connect: ' . mysql_error());
6 }
7
8 mysql_select_db("prueba", $con);
9
10 $q=mysql_query("SELECT * FROM quejas");
11
12 while($e=mysql_fetch_assoc($q))
13     $output[]=$e;
14
15 print(json_encode($output));
16
17
18 mysql_close($con);
19 ?>
```

Ilustración 25. Código PHP

Antes de analizar propiamente el código es necesario conocer unos conceptos básicos sobre PHP:

En PHP para comenzar y terminar el archivo se utiliza “<?php” y “?>” por ello todo lo que se encuentre entre estos símbolos será parte del código. Además se puede observar como hay algunas cadenas tienen el símbolo del dólar, todas las que lo poseen son variables.



Una vez entendido esto será más fácil comprender el código. Para empezar si se observa la primera línea se ve como hay un método “mysql\_connect”, que permite crear una conexión con el servidor de la base de datos pasándole la IP del servidor donde está alojada(“localhost”), el usuario(“android”) y la contraseña del usuario(“pass\_android”).

Esta conexión se almacenará en la variable \$con. Si ésta es incorrecta se mostrara el mensaje mysql\_error y si no a partir de “mysql\_select\_db” se selecciona la base de datos a la que queremos acceder, en este caso pasándole el nombre de la base de datos y la conexión establecida anteriormente. Una vez se ha accedido a la base de datos se construye la consulta con “mysql\_query” y se guarda en la variable \$q.

Una vez hecho esto, mientras que haya resultados para esa consulta se incluyen el *array* de salida (\$output), cuando ya hay no hay más resultados se codifica el *array* en *json* y se envía.

Posteriormente este *array* codificado en *json* se decodifica en la aplicación para obtener sus datos.

### DESCRIPCION DE LOS PHP

En este apartado se incluirá una tabla resumiendo el uso de cada PHP que se utiliza en la aplicación.

PHP	FUNCION
Insertarusuario.php	Inserta un usuario en la base de datos
Insertarfoto.php	Inserta una queja en la base de datos
Insertarapoyo.php	Inserta el apoyo de un usuario asociado a una queja en la base de datos
Insertarcomentario.php	Inserta el comentario realizado por un usuario sobre una queja en la base de datos
Contarapoyos.php	Cuenta el número de apoyos que tiene una determinada queja
Consultarusuario.nombre.php	Comprueba si un usuario está en la base de datos
Consultarusuario.php	Obtiene el correo de un determinado usuario
Consultarqueja.php	Obtiene las quejas correspondientes que cumplan con los campos introducidos
Consultarcomentario.php	Obtiene los comentarios relacionados con una queja
Consultageneral.php	Obtiene todas las quejas de la base de datos
Comprobarapoyo.php	Comprueba si un usuario ya ha apoyado una queja

Tabla 37. Descripción PHPs

## 4.2 APLICACIÓN ANDROID

En este apartado se mostrará toda la implementación de la aplicación Android y cómo se conecta con el servidor descrito anteriormente y los servicios web de Picasa y Google Maps.

### 4.2.1 INTRODUCCIÓN

Android es un sistema operativo móvil basado en Linux, que está presente en dispositivos móviles como teléfonos inteligentes, tabletas, Google TV y cada vez en más dispositivos como el proyecto de videoconsola Ouya. Es desarrollado por la Open Handset Alliance, la cual es liderada por Google pero en la cual también están presentes grandes empresas como Nvidia, Qualcomm, Samsung Electronics o las más recientes incorporadas Asustek, Toshiba y Vodafone.

La arquitectura de Android contiene varias capas que son de naturaleza bastante variada, como muestra la siguiente ilustración.



Ilustración 26. Capas Android



Empezando desde la capa más baja del sistema operativo se puede observar cómo para manejar los distintos dispositivos de hardware tiene un núcleo Linux, si se sube una capa más se puede observar la capa de las bibliotecas que están escritas en C y en C++ y el Android Runtime que contiene una máquina virtual Java (Dalvik Virtual Machine) que será la encargada de ejecutar las aplicaciones. En la siguiente capa hacia arriba se encuentran los distintos elementos y clases propios de la API y que serán utilizados por las aplicaciones. Por último, se encuentra con la capa de las aplicaciones las cuales están escritas en Java y cuyas componentes gráficas y de interfaz utilizan el lenguaje XML.

### *ESTRUCTURA DE ELEMENTOS EN UNA APLICACIÓN ANDROID*

Todas las aplicaciones Android siguen una misma estructura de archivos y carpetas independientemente de su tamaño y cada elemento cumple una serie de funciones que se explicarán a continuación:

- **AndroidManifest.xml:** Contiene una información esencial que el sistema debe tener antes de ejecutar la aplicación como:
  - Nombre del paquete java.
  - Descripción de los componentes de la aplicación.
  - Permisos.
  - Versión mínima del API de Android que necesita.
  - Dependencias de otras librerías.
- **Src:** Carpeta con el código fuente de la aplicación.
- **Assets:** Colección arbitraria de carpetas y ficheros.
- **Res:** Contiene los recursos de la aplicación: imágenes, iconos, menús, animaciones, etc.
- **android.jar:** Archivo “jar” que empaqueta las librerías Android para la plataforma correspondiente.
- **default.properties.xml:** Fichero que especifica las propiedades que serán usadas por el sistema de compilación.
- **Gen:** Carpeta con los .java generados automáticamente por las herramientas de Android.

## 4.2.2 ESTRUCTURA DE LA APLICACIÓN

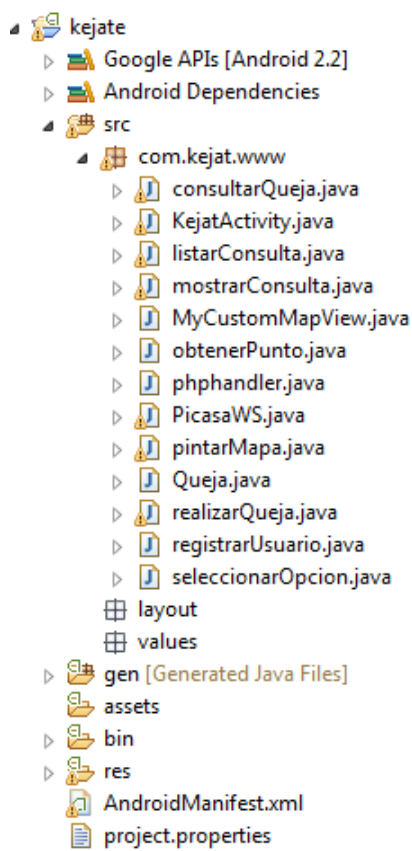


Ilustración 27. Estructura Aplicacion Android

Como se puede observar, en la carpeta “src” se encuentran los archivos “.java”. Cada uno se corresponderá con una actividad, un manejador o una clase Java que se ha creado para manejar la estructura de datos de la aplicación.

### KEJATACTIVITY.JAVA

Es la actividad principal de la aplicación y de la que se partirá hacia las demás, ésta se corresponde con el caso de uso de identificar usuario.

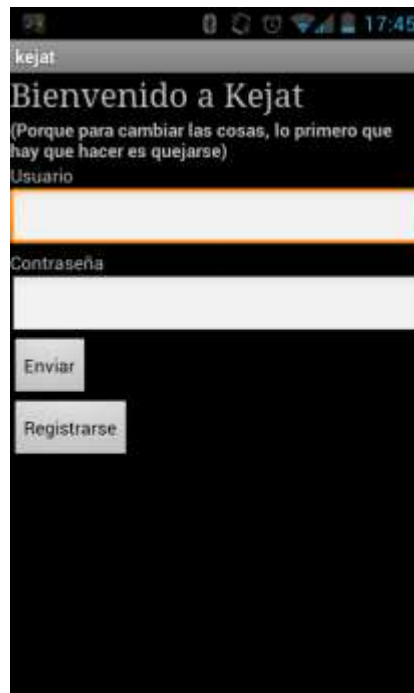


Ilustración 28. Pantalla Kejactivity

Como se puede observar en la imagen, en ella aparece el título, una frase motivadora para el usuario y dos cajas de introducción de texto en las cuales el usuario puede introducir su Nick y su contraseña y dos botones “enviar” y “registrarse”.

#### Descripción de la implementación del escenario básico:

- El usuario introduce su nombre y contraseña en las correspondientes cajas de texto.
- Éstas se guardan en unas variables de tipo string creadas en un objeto llamado SharedPreferences, el cual permite acceder al valor de estas variables en otras actividades distintas.
- Se crea un objeto a partir de la clase phphandler y se invoca a su correspondiente método “consultar” pasándole el archivo PHP “consultarnombreusuario.php” al que se quiere acceder y los campos introducidos, para consultar si el usuario está en la base de datos y obtener su dirección de correo. En caso de que el usuario no esté en la base de datos se le mostrará un mensaje que le indica al usuario que no está registrado y que se registre.
- Se crea un objeto de la clase PicasaWS, que pasándole el correo del usuario y la contraseña establece una conexión con Picasa si el usuario y el correo son correctos.

En caso de que el servicio de Picasa no permita acceder se le indicará al usuario que el nombre usuario o la contraseña son incorrectos.

- En el caso de que la verificación de la base de datos y de Picasa sea correcta se crea un *Intent* que lance la actividad seleccionarOpcion.
- Si el usuario pulsa el botón de registrarse en vez de en el de enviar, se crea un *Intent* que lanzará la actividad registrarUsuario.

### REGISTRARUSUARIO.JAVA

En esta actividad se registrarán los datos de un nuevo usuario en la base de datos.

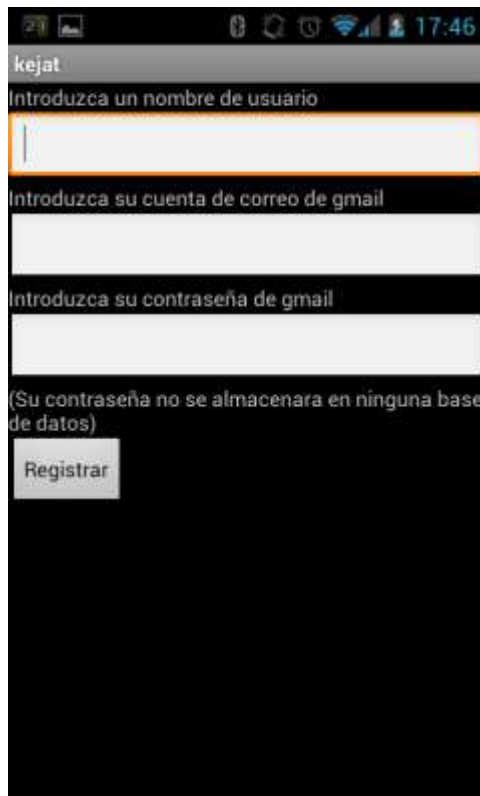


Ilustración 29. Pantalla Registrar usuario

Como se puede observar en la imagen, en esta actividad se muestran tres cajas de texto para que el usuario pueda introducir un *nick* o nombre de usuario, una cuenta de correo de Gmail y su correspondiente contraseña. Además se le indica con un mensaje que su contraseña no será almacenada en ninguna base de datos y un botón “registrarse”.





**Descripción de la implementación del escenario básico:**

- El usuario rellena los campos y pulsa en el botón “registrarse”.
- Se crea un objeto a partir de la clase phphandler, y se invoca a su correspondiente método pasándole el archivo PHP “consultarusuario.php” junto con los campos introducidos para consultar si el usuario está en la base de datos. En caso de que el usuario esté en la base de datos se le mostrará un mensaje que le indica que ya está registrado.
- Se crea un objeto de la clase PicasaWS que pasándole el correo del usuario y la contraseña nos establece una conexión con Picasa si ambos son correctos. En caso de que el servicio de Picasa no permita acceder se le indicará al usuario que el correo o la contraseña son incorrectos.
- En caso de que la verificación de la base de datos y de Picasa sea válida, se crea un objeto a partir de la clase phphandler, y se invoca a su método insertar pasándole el archivo PHP “insertarusuario.php” y los campos introducidos para que se inserte el usuario en la base de datos se muestra un mensaje al usuario de que el usuario se ha registrado correctamente.

### *SELECCIONAROPCION.JAVA*

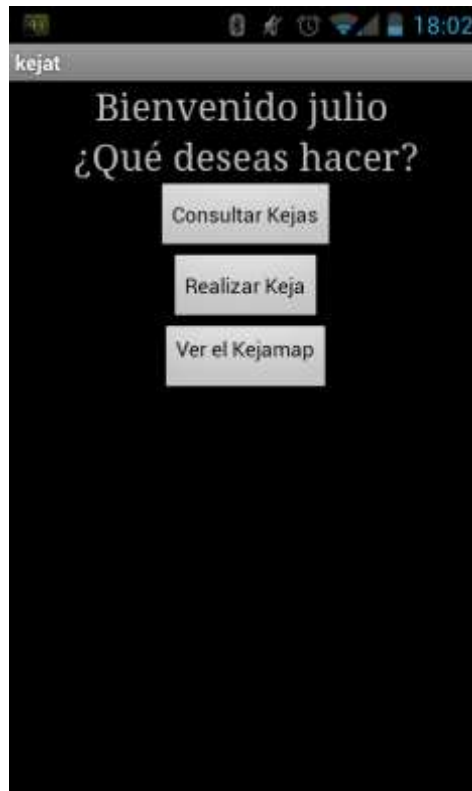


Ilustración 30. Pantalla Seleccionar opción

Esta actividad se encarga de analizar qué botón ha pulsado el usuario y de lanzar la actividad correspondiente.

Como se puede observar en la imagen se muestra al usuario un mensaje de bienvenida y se le da a elegir entre pulsar tres botones con las distintas acciones que puede realizar en la aplicación.

#### **Descripción de la implementación del escenario básico:**

- El usuario pulsa cualquiera de los distintos botones (Consultar Keja, Realizar Keja o Ver Kejamap) y se crea un objeto de la clase Intent que lanza la actividad correspondiente.

### *CONSULTARQUEJA.JAVA*

Esta actividad se encarga de recopilar los datos introducidos por el usuario sobre los que se quiere realizar la consulta.



**Ilustración 31. Pantalla Consultar queja**

Como se puede observar en la imagen, aparecen distintos elementos para que el usuario pueda rellenar los campos que desee. Dos de ellos son unos desplegables llamados “Spinners” los cuales tienen asociados varios “Strings” de manera que el usuario pueda seleccionarlos de manera sencilla e intuitiva. Además hay un botón que pone elegir fecha, que cuando se pulse aparecerá un “DatePicker”, que mostrará en pantalla la fecha actual y permitirá elegir otra pulsando unas flechas que tiene alrededor. Por último aparece el botón “Consultar”, que permitirá enviar la consulta al servidor.

#### **Descripción de la implementación del escenario básico:**

- El usuario rellena los distintos campos sobre los que desea realizar la consulta y pulsa en el botón “Consultar”.

- Si el usuario ha seleccionado algún valor para el radio se llamará a la función GetGPS que accede al gps del teléfono y devuelve su posición en coordenadas geográficas.
- Una vez recopilados todos los datos que ha introducido el usuario, se crea un Intent encargado de lanzar la actividad “listarConsulta” y se le añade los datos obtenidos del usuario para que puedan ser utilizados por ella.

### LISTARCONSULTA.JAVA

Esta actividad pertenece a una subclase de la clase Activity llamada ActivityList la cual indica que esta actividad va a ser la encargada de mostrar una lista de objetos.



Ilustración 32. Pantalla Insertar queja

Como se puede observar para esta lista (ArrayList) se asociará un *array* con las quejas obtenidas de la consulta y una vista especial para que muestre una miniatura de las quejas y parte de su información. Además se puede observar en la parte superior que aparecen un botón “+” y otro “-” y un número que indica el número de página actual.

#### Descripción de la implementación del escenario básico:

- Desde la actividad principal se invoca al método `getItems()`, el cual crea un objeto a partir de la clase `phphandler` y se invoca a su correspondiente método `consultar`, pasándole el archivo PHP (`consultarqueja.php`) al que se quiere acceder y los campos

introducidos en la actividad anterior. Los resultados obtenidos los mete en un array de quejas y este se devuelve a la actividad principal.

- El array obtenido se le pasa a la función rellenar() la cual se encarga de trocear el array en páginas dependiendo de los resultados que se quieran mostrar por página (en este caso 5).
- Se asocia este array a la vista del del arraylist y se le muestra al usuario.
- Si el usuario pulsa sobre alguna de las quejas de la lista se crea un nuevo Intent con los datos de la queja seleccionada y se lanza la actividad mostrarConsulta.

## *MOSTRARCONSULTA.JAVA*

Esta actividad se encarga de mostrar los datos de la queja seleccionada y las distintas operaciones que se pueden realizar sobre ella.

Como se muestra en la imagen, se aprecian dos partes. La primera parte contiene los datos de la propia queja, un texto que indica el número de apoyos, un botón para apoyar la queja, otro para realizar un comentario y otro para compartir la queja con otras personas. En la segunda parte se muestra una lista con todos los comentarios realizados sobre la queja.



Ilustración 33. Pantalla Mostrar consulta

### Descripción de la implementación del escenario básico:

- Primero se obtienen del Intent los datos pasados en la actividad anterior.
- Se muestran los datos incluyendo la imagen descargada de la correspondiente *url* a través de la función “descargarimagen”.
- Se crea un objeto a partir de la clase “phphandler”, y se invoca a su correspondiente método *consultar* pasándole el archivo PHP “comprobarapoyo.php”, el *nick* de usuario y la *url* de la imagen; para comprobar si el usuario ya ha apoyado esa queja. En caso afirmativo no se mostrará el botón de apoyar y se mostrará un texto que diga “a ti y a X personas más no les gusta que pase esto”. En el caso de que el usuario no haya votado se le mostrará el botón para que lo pueda hacer.
- Se crea un objeto a partir de la clase “phphandler”, y se invoca a su correspondiente método “*contar*” pasándole el archivo PHP “contarapoyos.php” y la *url* de la imagen. En el caso de que sea distinto de cero se muestra el mensaje anterior pero se cambia la “X” por el número de apoyos obtenidos.
- Si el usuario pulsa el botón “no me gusta”, se crea un objeto a partir de la clase *phphandler*, y se invoca a su correspondiente método *insertar* pasándole el archivo PHP “contarapoyos.php”, la *url* de la imagen y el nombre del usuario. Para insertarlo en la tabla de apoyos, una vez hecho esto se actualiza el mensaje anterior incluyendo “y a ti” y se deja de mostrar el botón “no me gusta”.
- Si el usuario pincha en el botón “añadir comentario” aparece una caja de texto y un botón “enviar”. Una vez creado el usuario escribe el comentario y pulsa el botón “enviar”, se crea un objeto a partir de la clase “phphandler”, y se invoca a su correspondiente método *insertar* pasándole el archivo PHP “insertarapoyo.php”, la *url* de la imagen y el nombre del usuario. Cuando se inserta la tabla además se añade la fecha y la hora a la que se ha realizado la operación.
- Si el usuario pincha en el botón “compartir” se lanza un Intent que lanzara por defecto la actividad de compartir del sistema operativo y que permitirá compartir la queja en distintas plataformas en función de los programas que se tengan instalados en el móvil.
- En cuanto a la lista de comentarios se trata de manera similar a la de actividad “listarConsulta” pero con objetos “comentarios” en vez de “quejas”.

### REALIZARQUEJA.JAVA

Esta actividad se encargará de recopilar los datos referentes a una queja del usuario e insertarlos en la base de datos.



Ilustración 34. Pantalla Realizar queja

Como se puede observar en la imagen se disponen de dos cajas de texto para que el usuario introduzca el título y la descripción de la queja, un *spinner* para elegir el tipo, y tres botones. Dos de ellos “Seleccionar foto del móvil” y otro “Modificar dirección” que lanzaran sus correspondientes subactividades.

#### Descripción de la implementación del escenario básico:

- El usuario rellena todos los campos y pulsa en “seleccionar foto del móvil”, al hacer esto se crea un Intent que lanza una subactividad por defecto que permite acceder a la galería de imágenes. Una vez el usuario selecciona una de las imágenes se llama a la función “getpath”, que nos permite acceder a la ruta de la imagen, esta se guarda en un array de bits y se procede a analizar sus cabeceras EXIF para ver si contiene información geográfica sobre la imagen, si es así esta se guarda en las variables longitud y latitud. Y se devuelven los valores a la actividad principal.



- Si el usuario pulsa en “modificar dirección” se crea un Intent que lanzará la subactividad referente al archivo “obtenerPunto.java”, la cual muestra un mapa obtenido de GoogleMaps, para ello se requiere incluir en el XML una licencia que se mostrará cómo obtenerla en uno de los anexos. En este mapa se le permite al usuario buscar una dirección y una vez se ha encontrado si el usuario realiza una pulsación larga sobre ella, se obtiene la longitud y la latitud y se devuelve a la actividad principal.
- Si el usuario pulsa en el botón enviar se comprueba que todos los campos estén rellenos y en caso contrario se muestra un mensaje al usuario indicándoselo. Una vez con todos los campos rellenos se crea un objeto de la clase “PicasaWS” a partir del correo y la contraseña del usuario, que permite establecer una conexión con el servicio web de Picasa. Posteriormente se llama al método “addPicture” del objeto “PicasaWs” pasándole la imagen y un nombre de álbum, este a su vez, si el álbum no está creado llama a el método createalbum de la clase PicasaWS, que crea al álbum. Una vez el creado el álbum se añade la imagen a éste y se analiza a la respuesta de servicio web Picasa para obtener la url de la imagen.
- Una vez se tiene la *url* de la imagen se pasa junto con los demás campos y el PHP (insertarfot.php) para crear un objeto de la clase “phphandler”, después se llama al método insertar del objeto, este se inserta en la base de datos y se muestra un mensaje al usuario de que la queja se ha registrado correctamente.



### *PINTARMAPA.JAVA*

Esta actividad se encarga de mostrar un mapa con todos los puntos en los cuales encuentra una queja de algún ciudadano.



Ilustración 35. Pantalla Pintar mapa

Como se puede observar en la imagen cada uno de los iconos muestra la posición de la queja de un usuario.

#### **Descripción de la implementación del escenario básico:**

- Primeramente la actividad crea un objeto phphandler a partir del archivo (consultageneral.php) y se llama al método consular de éste. Con los resultados obtenidos se crea una lista objetos “quejas”, se crea una capa encima del mapa y a partir de la lista de quejas se coloca cada elemento en su correspondiente posición geográfica.



- Si el usuario pulsa sobre el icono de una queja se crea un Intent incluyendo los datos de la queja correspondiente y se lanza la actividad perteneciente al archivo “mostrarConsulta.java” descrita anteriormente.

### *QUEJA.JAVA*

Este es uno de los archivos que no contiene ninguna actividad, sino que ha sido creado para manejar nuestra clase “queja”, básicamente se compone de un constructor y una serie de métodos los cuales permiten acceder a estos sin romper el encapsulamiento de la clase.

#### **Atributos:**

- Titulo: Es de tipo String y contiene el título de la queja.
- Usuario: Es de tipo String y contiene el nombre del usuario que ha realizado la queja.
- Foto: Es de tipo String y contiene la url de la imagen alojada en Picasa
- Descripcion: Es de tipo String y contiene una breve descripción de la queja.
- Tipo: Es de tipo String e indica la naturaleza la de la queja.
- Latitud: Es de tipo Double y contiene la latitud correspondiente a la posición geográfica de la queja.
- Longitud: Es de tipo Double y contiene la longitud correspondiente a la posición geográfica de la queja.
- Fecha: Es de tipo String y contiene la fecha en la que se realizo la fotografía de la queja.

### *PHPHANDLER.JAVA*

Este archivo es una clase de “manejador” que permite establecer una conexión con la base de datos a partir de los archivos PHP alojados en el servidor. Posee tres métodos los cuales permiten realizar operaciones de contabilización, consulta e inserción sobre la base de datos, mediante la utilización de métodos http POST.

### *PICASAWS.JAVA*

Este archivo es otro de los manejadores que en este caso permite acceder al servicio web de Picasa, mediante operaciones http GET y http POST. Principalmente posee tres métodos:

- El constructor, el cual a partir de él y pasándole el correo del usuario y la contraseña permite acceder al servicio web.
- Createalbum: Este permite crear un álbum en el servicio web de Picasa con el nombre que recibe.
- AddPicture: Permite añadir una imagen a un determinado álbum.

## 4.3 GOOGLE MAPS

La utilización de Google Maps está tan integrada en la plataforma Android que cuando se quiere crear un proyecto, da la posibilidad de utilizar sólo la API de Android o una API de Google, la cual incluye la API de Google Maps, tal y como se puede apreciar en la imagen.

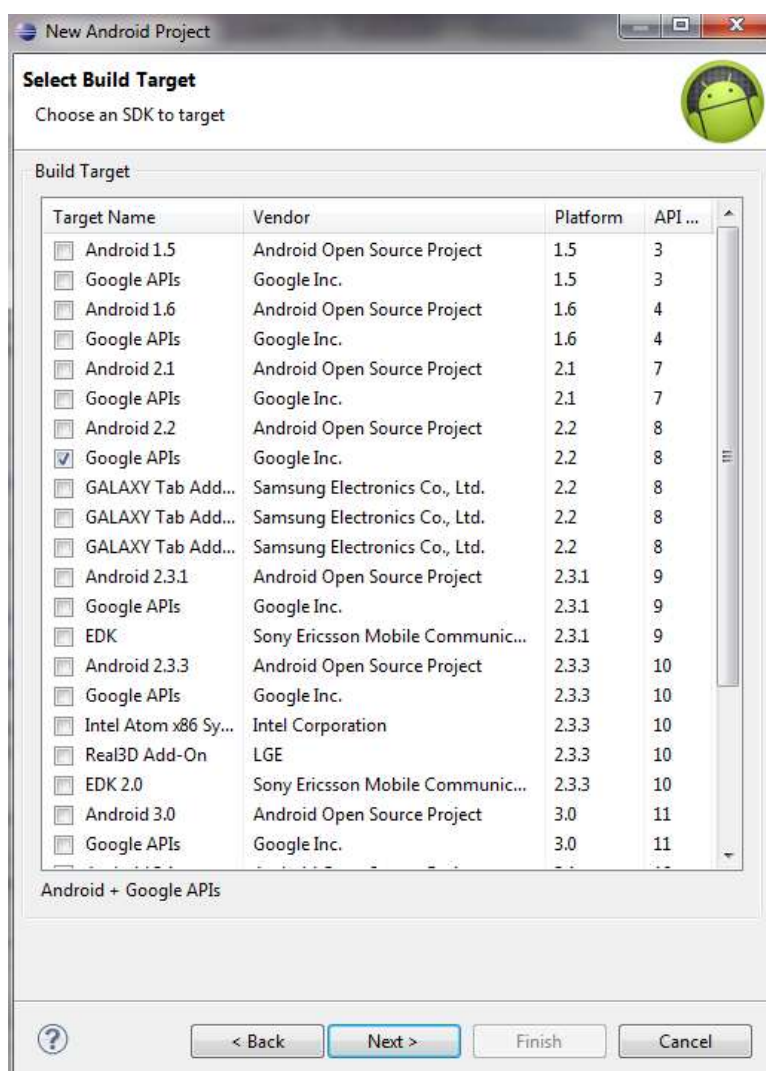


Ilustración 36. APIS de Android

Por lo demás cuando se quiera realizar una actividad que utilice un mapa de Google Maps se procederá como otra cualquiera, lo único que la actividad será del subtipo MapActivity y la vista de tipo Mapview.

Otro aspecto que hay que tener en cuenta para la implementación de la parte de los mapas es que para poder utilizarlos se necesita una clave, llamada “apiKey”, que proporcionará Google y que se tiene incluir en los xml que utilicen los mapas. El proceso de obtención de esta clave se desarrollara en uno de los anexos.



```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent" android:layout_height="fill_parent">
    <com.kejat.www.MyCustomMapView
        android:id="@+id/mapView" android:layout_width="fill_parent"
        android:layout_height="fill_parent" android:clickable="true"
        android:enabled="true" android:apiKey="05k[REDACTED]_Mw" />

    <LinearLayout android:id="@+id/searchPanel"
        android:layout_width="fill_parent" android:layout_height="wrap_content"
        android:layout_alignParentTop="true" android:layout_centerHorizontal="true"
        android:visibility="invisible">
        <EditText android:id="@+id/searchText" android:layout_width="fill_parent"
            android:layout_height="wrap_content" android:layout_weight="1" />
        <Button android:id="@+id/searchButton" android:layout_width="wrap_content"
            android:layout_height="wrap_content" android:text="Search" />
    </LinearLayout>
</RelativeLayout>
```

Ilustración 37. Código con Apikey de Google Maps

## 4.4 PICASA WEB SERVICE

La API para acceder al servicio web de Picasa se encuentra disponible para desarrollar el acceso en varios lenguajes y hay una cantidad bastante extensa de documentación [13]. En ella existe un apartado especial con bibliotecas sólo para Java y que permite tratar las conexiones, los álbumes y fotografías como objetos propios de Java. Sin embargo cuando se empezó este proyecto estas bibliotecas no funcionaban de manera muy estable, por lo tanto se decidió seguir la guía de protocolo para desarrolladores, realizando conexiones mediante operaciones de GET y POST contra el servicio web, de manera similar a la implementada en el acceso a los PHP del servidor, como se puede observar en las clases phphandler.java y PicasaWs.java.



## CAPÍTULO 5. EXPERIMENTAL

### 5.1 INTRODUCCIÓN

En este capítulo se describirá las simulaciones de datos, las pruebas realizadas, así como los elementos utilizados para llevarlas a cabo. Antes de empezar a describir las distintas partes de este capítulo, cabe mencionar que algunos tipos de pruebas se limitaran a la verificación de los controles de errores y a la descripción de su funcionamiento; ya que esta aplicación está desarrollada con la intención de que el usuario no pueda cometer errores en su manejo.

### 5.2 PRUEBA DE RENDIMIENTO

Para llevar a cabo esta prueba se analizará la operación que más tiempo consumen en la aplicación, la de realizar queja.

La prueba que se llevará a cabo se realizará con un dos móviles (Samsung Galaxy S y Samsung Galaxy Ace), un línea de fibra óptica de 50 Mb de Movistar con un router inalámbrico modelo Comtrend V2 11n FTTH y un servidor Mowes Portable sobre un ordenador portátil hp Pavilion dv6. Todas las especificaciones técnicas de los recursos utilizados se mostrarán posteriormente en sus correspondientes anexos.

#### 5.2.1 PRUEBA DE RENDIMIENTO PARA REALIZAR QUEJA

La prueba consistirá en tomar 10 tiempos de lo que tarda en realizarse la operación en cada uno de los móviles, se hará la media y se compararan los resultados. Para la realización de esta prueba se utilizará una fotografía geoetiquetada de 1280x960 y cuyo peso es de 0,97 MB.

Una vez se han introducido todos los campos se pulsa el botón enviar y se miden los tiempos, para la subida de la imagen a Picasa, para la inserción en la base de datos y el tiempo total. Según estas condiciones se han obtenido los siguientes datos.

Intento	Tiempo Total	Tiempo Picasa	Tiempo Servidor
1	14792	14750	42
2	6791	6721	70
3	7580	7545	35
4	6506	6473	33
5	11522	11490	32
6	11910	11879	31
7	8532	8462	70
8	6499	6429	70
9	6276	6245	31
10	6479	6404	75

Tabla 39. Tiempos Samsung Galaxy Ace

Intento	Tiempo Total	Tiempo Picasa	Tiempo Servidor
1	10742	10576	166
2	2722	2692	30
3	3323	3241	82
4	2518	2474	44
5	7118	7054	64
6	7674	7643	31
7	5470	5411	59
8	5801	5646	155
9	2564	2538	26
10	7963	7896	67

Tabla 38. Tiempos Samsung Galaxy S

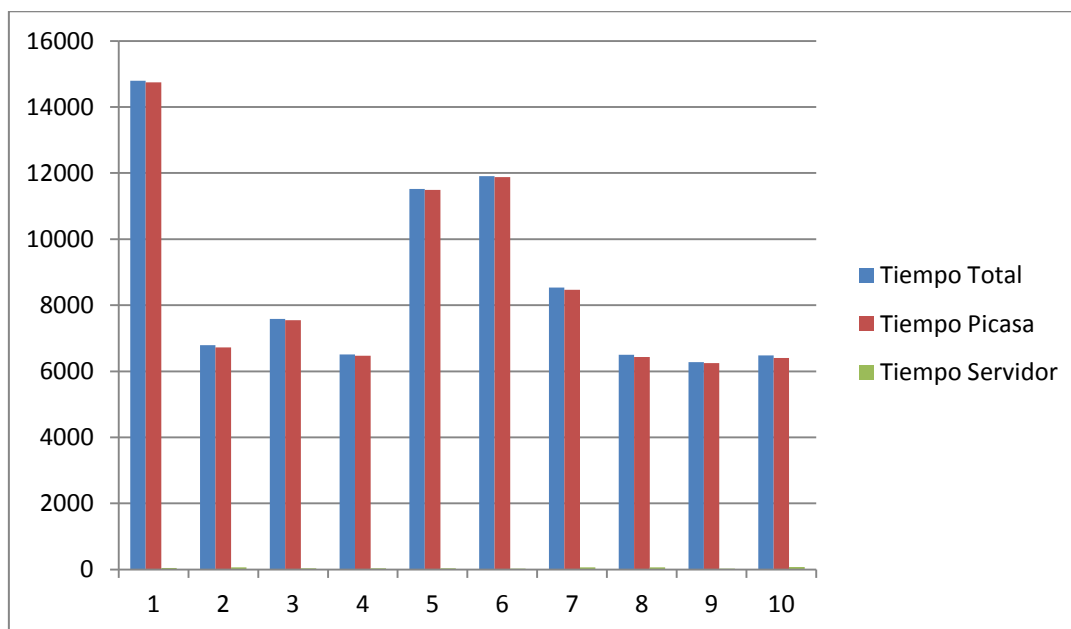


Ilustración 38. Gráfico Samsung Galaxy Ace

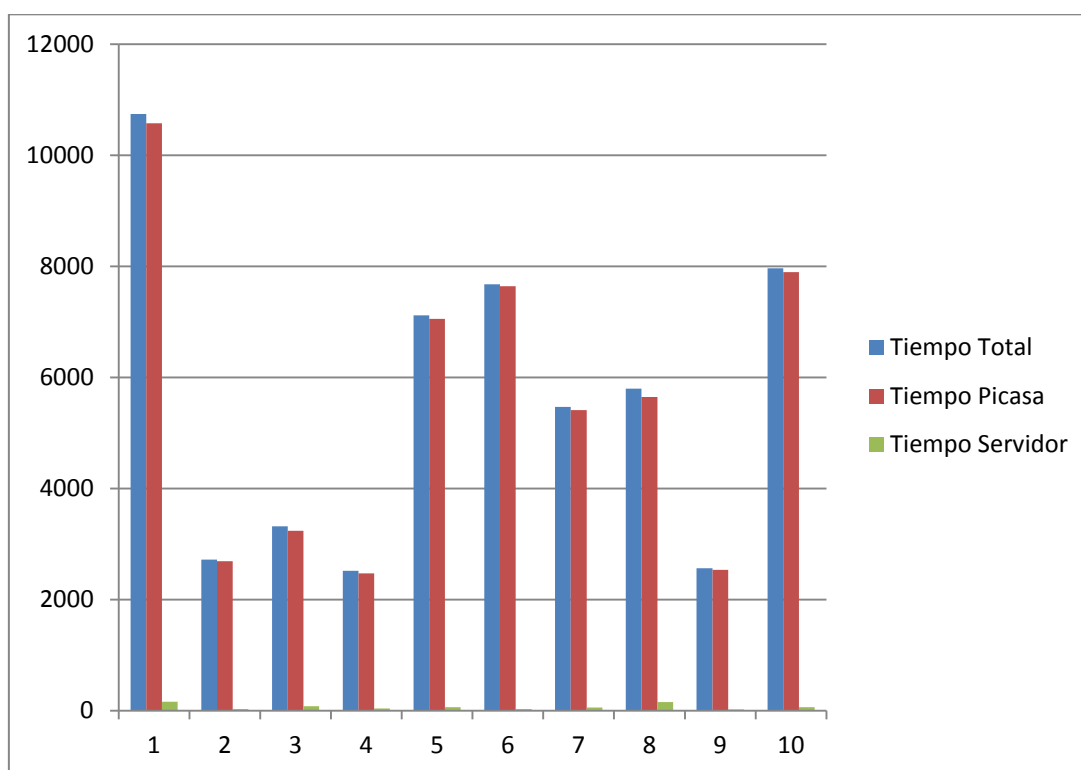


Ilustración 39. Gráfico Samsung Galaxy S

A la vista de estos datos se pueden observar varias cosas:

- En las dos tablas el primer intento siempre es el que mayor tiempo tarda en realizar la operación, esto se debe a que todavía tiene operaciones pendientes de la toma de datos, y para las posteriores, como ya están introducidos, el sistema operativo prevé la operación que se va a realizar cuando se pulse en “enviar”.
- Se puede observar variaciones bastante grandes de un intento a otro. Esto es debido a que los tiempos no son medidos en tiempo de ejecución del procesador, ya que hay que conectarse a un servicio web externo como es el de Picasa. Como el tiempo de respuesta de Picasa es variable de ahí esas diferencias tan grandes.
- El tiempo de inserción de la queja en la base de datos es prácticamente despreciable con respecto al de la subida de la imagen a Picasa, esto es debido a que el intercambio de datos es mucho mayor en Picasa (el tamaño de la foto) y a que el servidor se encuentra en nuestra misma red WiFi por lo tanto el tiempo de acceso es menor.
- Aunque la diferencias de tiempos es bastante grande de un intento a otro si se hace la media de cada dispositivo móvil se obtiene 8688,7 ms para el Samsung Galaxy Ace y 5589,5 ms para el Samsung Galaxy S. Este resultado es lógico debido a que el Samsung Galaxy S es de una gama mayor y tiene mayor capacidad de procesamiento como se puede observar en las especificaciones técnicas o en las referencias [32] y [33].

### 5.3 PRUEBAS DE INTRODUCCIÓN DE DATOS

En este apartado se analizarán las tres tipos de pruebas de introducción de datos y como se comporta la aplicación en ellas. Las pruebas serán de desbordamiento, de valores nulos y de coherencia de datos.

#### 5.3.1 PRUEBAS DE INTRODUCCIÓN DE DATOS (DESBORDAMIENTO)

Como ya se ha mostrado anteriormente en el apartado de implementación de la base de datos, cada campo tiene una longitud fija. En la imagen que se muestra a continuación se muestra la longitud de los campos para que se correspondan con la base de datos.

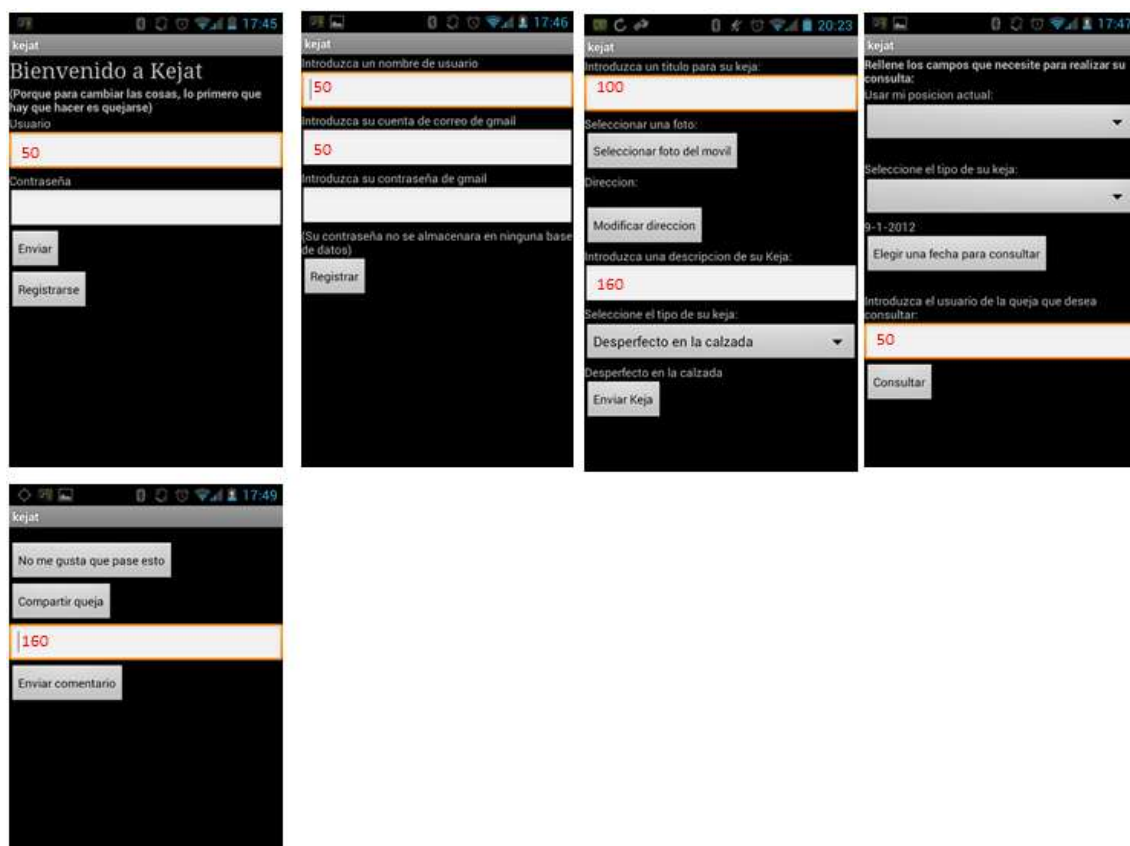


Ilustración 40. Tamaño maximo de campos



En el caso de que se introdujera una cadena mayor que esa longitud, a la hora de insertar esa cadena en la base de datos daría un fallo y no se insertarían los datos. Para que no pase esto la API de Android permite poner una capacidad máxima a los TextEdit (campos de texto).

Como en este proyecto se ha incluido la longitud máxima, esta prueba únicamente se dedicará a comprobar que en todos los campos de texto no deja introducir más caracteres que la longitud máxima y que esta se corresponde con la base de datos.

Una vez realizada esta prueba se puede concluir que la aplicación se comporta de manera correcta y que cumple con lo anteriormente mencionado.

## 5.3.2 PRUEBAS DE VALORES NULOS

En este apartado se analizarán la introducción de los distintos valores nulos y la influencia en el funcionamiento de la aplicación. Para ello, como no en todos los campos será la misma, se procederá a enumerar todas las entradas de datos.

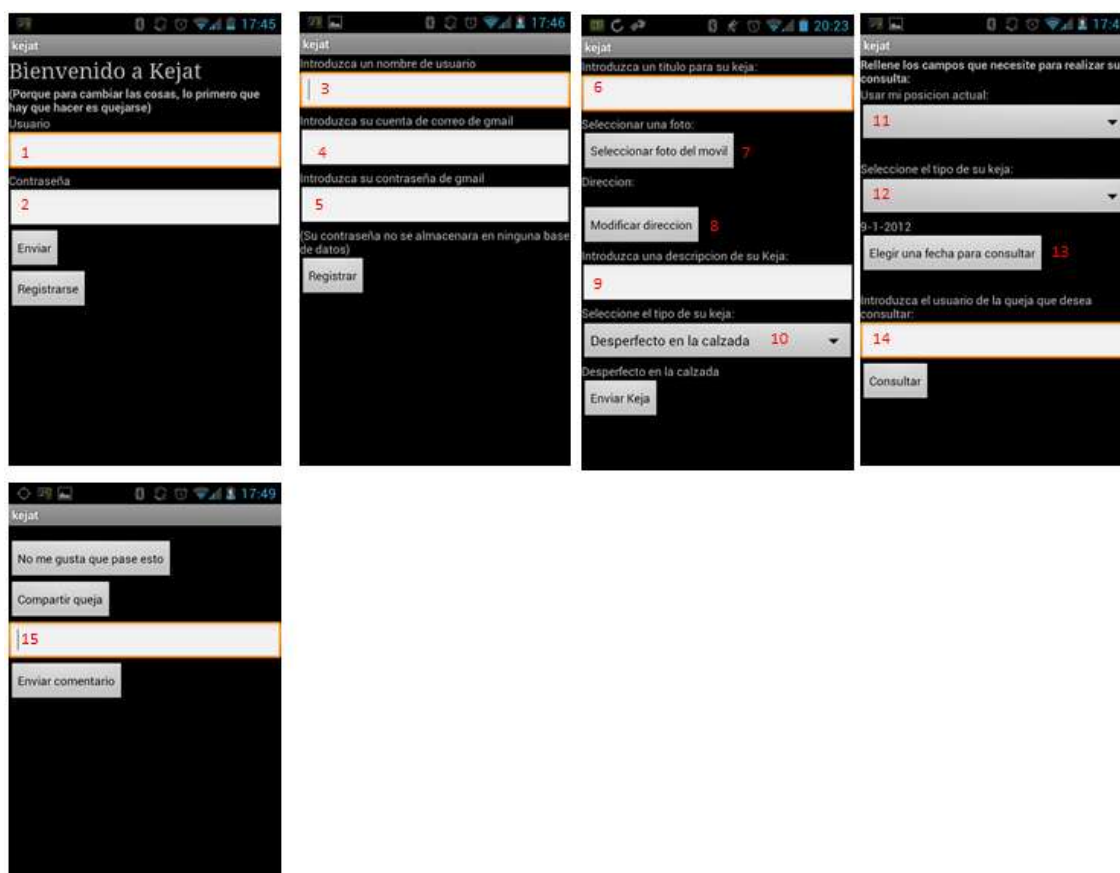


Ilustración 41. Pantallas de Introducción de datos



Una vez enumeradas las entradas de los datos se procederá a analizar cada una de ellas:

### *Entradas 1(usuario) y 2(contraseña):*

Se procederá a analizar las dos en conjunto debido a que una complementa a la otra y las dos tienen que tener concordancia entre ellas. Una vez explicado esto estos son los distintos casos:

- **Usuario nulo y contraseña nula:** El sistema muestra dos mensajes, el primero que el usuario no está registrado y el segundo que la contraseña o el usuario son incorrectos.
- **Usuario registrado y contraseña nula:** El sistema muestra un mensaje de que la contraseña o el usuario son incorrectos.
- **Usuario nulo y contraseña no nula:** El sistema muestra dos mensajes, el primero que el usuario no está registrado y el segundo que la contraseña o el usuario son incorrectos.

### *Entradas 3(usuario), 4(correo), 5(contraseña)*

Estas entradas también se analizarán en conjuntos por la misma razón que en el apartado anterior.

- **Usuario nulo, correo nulo, contraseña nula:** En este caso el sistema muestra dos mensajes, el usuario no está registrado y la contraseña introducida es incorrecta.
- **Usuario nulo, correo nulo, contraseña no nula:** En este caso el sistema muestra los mismos mensajes que en el caso anterior.
- **Usuario nulo, correo no nulo, contraseña nula:** El sistema sigue mostrando el mismo mensaje que en los casos anteriores.
- **Usuario nulo, correo no nulo, contraseña no nula:** En este caso si el correo y la contraseña son verdaderas crearía un usuario cuyo identificador sería “ ”.
- **Usuario no nulo, correo nulo, contraseña nula:** En este caso mostraría los mensajes de los casos anteriores.
- **Usuario no nulo, correo nulo, contraseña no nula:** En este caso mostraría los mensajes de los casos anteriores.
- **Usuario no nulo, correo no nulo, contraseña nula:** En este caso si el usuario y el correo ya están registrados mostraría un mensaje de que la contraseña es incorrecta y que el usuario ya está registrado, en caso contrario mostraría los mismos mensajes que en los casos anteriores.
- **Usuario no nulo, correo no nulo, contraseña no nula:** En este caso si los datos son concordantes y el usuario está registrado mostraría un mensaje de el usuario ya está registrado y si el usuario no está registrado mostraría un primer mensaje de que el usuario no está registrado y posteriormente que el usuario se ha registrado



correctamente. En cualquier otro caso se mostrarían los mensajes de los apartados anteriores.

### ***Entradas de la 6 a la 10***

En este caso las entradas se corresponden con los campos de la tabla queja: titulo, foto, latitud, longitud, descripción, tipo y fecha.

Para los casos 6, 9 y 10 si se introducen valores nulos, el sistema mostrará un mensaje pidiendo por favor que se introduzca el campo correspondiente. Por ejemplo, si se introduce el valor nulo en la entrada 6 el sistema mostrará un mensaje “Por favor introduzca un título para su queja”.

Los campos de latitud, longitud, fecha y foto se obtienen de la imagen seleccionada a partir de seleccionar foto del móvil, en caso de que la imagen no está geoetiquetada se le mostrará un mensaje al usuario de que la imagen no tiene información geográfica y por favor seleccione una dirección.

### ***ENTRADAS DE LA 11 A LA 14***

En este caso como la operación va a ser de consulta, sí se va permitir al usuario introducir valores nulos. Esto querrá decir que el usuario no querrá realizar la búsqueda sobre el campo seleccionado.

Esto que a priori puede parecer bastante sencillo se deberá controlar en el PHP correspondiente ya que si no se tuviera en cuenta la consulta se realizaría con los campos=null.

Por ejemplo:

En este ejemplo el usuario quiere ver todas las quejas que ha realizado por lo tanto rellenaría el formulario de búsqueda de la siguiente manera:



Ilustración 42. Caso Consultar queja

En este caso sería bastante simple la consulta que quiere hacer el usuario, la traducción a MySQL de lo que quiere hacer sería así:

```
SELECT * FROM quejas WHERE usuario='nombre del usuario';
```

Pero lo que el sistema interpretaría si no se controlara sería:

```
SELECT * FROM quejas WHERE (usuario='nombre del usuario' AND latitud=NULL AND  
longitud=NULL AND radio=NULL AND tipo=NULL AND fecha=NULL);
```

Es decir el sistema buscaría dentro de la base de datos todos los registros que coincidieran con el usuario=nombre del usuario pero además que también tuvieran los demás campos de la consulta=NULL. Como al realizar la queja todos los campos son obligatorios, no habría ningún registro que tuviera ningún campo=null. Por lo tanto el sistema no mostraría ningún resultado y el usuario se quedaría sin ver sus quejas.

Este problema se ha solucionado analizando en el PHP cuales de los valores recibidos son no nulos y una vez hecho esto se procede a la construcción de la consulta y a su posterior ejecución.

### **ENTRADA 15 (comentario)**

Esta entrada se encuentra en la clase mostrarQueja.java y sirve para que el usuario pueda escribir un comentario sobre una denuncia, que posteriormente se almacenará en la base de datos. Por lo tanto, para conservar la consistencia de la base de datos se podrá permitir que contenga un valor nulo.



Para controlar esto se verifica si es un valor nulo o no antes de mandarlo al servidor, en caso de que lo sea no se manda y se le muestra un mensaje al usuario.

Si se ejecuta la aplicación y se introduce un valor nulo se puede observar como el sistema muestra un mensaje de que introduzca algo en el comentario. Por lo tanto se puede concluir que en este aspecto se comporta de manera correcta la aplicación.

## 5.4 PRUEBAS DE INTERGRACIÓN CON PICASA

En este apartado se describirán las pruebas realizadas y las conclusiones obtenidas de las pruebas realizadas con el servicio web de Picasa, aunque la API de Picasa es muy extensa estas pruebas solo se centrarán en distintos escenarios que pueden afectar al funcionamiento del sistema.

### 5.4.1 PRUEBA DE DESBORDAMIENTO

Tras realizar distintas pruebas de simulación se ha llegado a una conclusión que no está descrita en los límites de la cuota de almacenamiento de Picasa. Esta es que en un mismo álbum no se pueden almacenar más de 1000 imágenes, en este caso nuestro sistema fallaría y se cerraría la aplicación. Se ha decidido no controlarlo debido a que es un caso extremo y que prácticamente ningún usuario que haga un buen uso de la aplicación llegaría a este límite y en caso de que se produjera se le proporcionaría una cuenta nueva facilitada por el equipo y que representaría que es un usuario VIP de la aplicación.

### 5.4.2 PRUEBA DE ACCESO A IMÁGENES

En este caso la prueba consistirá en el acceso a una imagen que está en la base de datos pero que ya ha sido borrada en Picasa. Este caso puede producirse varias veces debido a que las imágenes se almacenan en Picasa es a través de un álbum de la cuenta del usuario y como propietario que es de su cuenta puede acceder directamente al servicio web de Picasa y borrar una determinada fotografía.

Para la realización de esta prueba se realiza una queja, se accede a Picasa desde el servicio web y se borra la imagen. Una vez hecho esto la imagen no se encontrará en Picasa pero si en la base de datos del sistema. Se procede a la consulta de esa queja y se puede observar como la aplicación no falla y sigue comportándose de manera normal pero



mostrando el icono de la aplicación en vez de la imagen ya borrada. Por lo tanto se puede concluir que el resultado de esta prueba ha resultado satisfactorio.

### 5.4.3 PRUEBAS DE ACCESO CON DISTINTAS CUENTAS

Tras muchas pruebas realizadas con muchas cuentas, se ha llegado a la conclusión de que, aunque si bien según Google se tiene acceso a Picasa sólo por tener una cuenta *gmail*, es necesaria la activación de un perfil de Google para asegurarse que la aplicación no va a dar errores. En caso contrario es posible que Google deniegue el acceso a Picasa a través de su API y cuando se quiera realizar una queja falle la aplicación.



## CAPÍTULO 6. ORGANIZACIÓN DEL PROYECTO

### 6.1 ORGANIZACIÓN Y COMPETENCIAS DEL PERSONAL

Para que este proyecto se pueda llevar a cabo a gran escala por una empresa, se ha de tener en cuenta qué recursos de personal son necesarios. Por ello en este apartado se establecerán los niveles jerárquicos dentro del proyecto, así como las competencias y responsabilidades de cada uno de ellos.

#### 6.1.1 JEFE DE PROYECTO

Es el máximo responsable del proyecto ante la empresa, encargado de supervisar y dirigir en todo momento el proyecto y el grupo de trabajo. En cuanto a sus competencias son las siguientes:

- Diseñar y organizar los planes de calidad, gestión de riesgos y gestión de la configuración.
- Diseñar la aplicación.
- Asignar los recursos necesarios para cada una de las fases del proyecto.
- Tener capacidad de reacción ante cualquier cambio que se produzca en el desarrollo del proyecto.
- Supervisar cada una de las fases del proyecto.
- Asegurarse del cumplimiento de los plazos asignados al proyecto.

#### 6.1.2 ANALISTA

Es la mano derecha del jefe de proyecto y el encargado de documentar todo lo que se realice durante el desarrollo del proyecto. En cuanto a sus competencias son las siguientes:



- Ayudar al jefe de proyecto en el diseño y organización los planes de calidad, gestión de riesgos y gestión de la configuración.
- Ayudar en el diseño de la aplicación.
- Realizar el seguimiento de los planes de calidad, gestión de riesgos y gestión de la configuración.
- Documentar todas las fases del proyecto.

### 6.1.3 PROGRAMADOR

Es la persona que hace que transforma el proyecto en algo tangible. Sus competencias son las siguientes:

- Participar en el diseño de la aplicación.
- Codificar la aplicación.
- Realizar pruebas a la aplicación.
- Terminar su trabajo en el tiempo establecido

## 6.2 PLANIFICACIÓN DEL PROYECTO

En este apartado se describirá la planificación del proyecto a partir de un diagrama de Gantt y sus tareas correspondientes. A continuación se mostrará la tabla de tareas del proyecto.

### 6.2.1 TABLA DE TAREAS

Nombre de tarea	Duración	Comienzo	Fin
Inicio proyecto	0 días	jue 01/09/11	jue 01/09/11
<b>1 Planificación y Requisitos</b>	<b>16,54 días</b>	<b>jue 01/09/11</b>	<b>vie 23/09/11</b>
<b>1.1 Inicio y Establecimiento de requisitos</b>	<b>16,54 días</b>	<b>jue 01/09/11</b>	<b>vie 23/09/11</b>
1.1.1 Identificar ideas y necesidades	1,38 días	jue 01/09/11	vie 02/09/11
1.1.2 Formular posibles aproximaciones a la solución	5,25 días	vie 02/09/11	lun 12/09/11
1.1.3 Realizar estudio de viabilidad	2,75 días	lun 12/09/11	mié 14/09/11
<b>1.1.4 Establecimiento de Requisitos</b>	<b>6,67 días</b>	<b>mié 14/09/11</b>	<b>vie 23/09/11</b>
1.1.4.1 Establecer requisitos de usuario	4,17 días	mié 14/09/11	mié 21/09/11
1.1.4.2 Establecer requisitos del software	4,17 días	mié 14/09/11	mié 21/09/11
1.1.3.3 Establecer los requisitos de interfaz	6,17 días	mié 14/09/11	vie 23/09/11
<b>1.2 Planificación</b>	<b>11,13 días</b>	<b>vie 02/09/11</b>	<b>lun 19/09/11</b>



1.2.1 Planificación del proyecto y asignación de los recursos	8,75 días	vie 02/09/11	lun 19/09/11
<b>1.4 Documentación</b>	<b>6,88 días</b>	<b>mié 07/09/11</b>	<b>jue 15/09/11</b>
<b>2 Diseño</b>	<b>32,42 días</b>	<b>vie 23/09/11</b>	<b>mar 08/11/11</b>
2.1 Diseño detallado de la arquitectura del sistema	5 días	vie 23/09/11	lun 03/10/11
2.2 Diseño de la base de datos	7 días	vie 23/09/11	mié 05/10/11
2.3 Diseño de interfaces	10 días	vie 23/09/11	lun 10/10/11
2.4 Diseño de la aplicación Android	15 días	lun 10/10/11	lun 31/10/11
2.5 Diseño detallado del sistema	6,92 días	lun 31/10/11	mar 08/11/11
<b>2.7 Documentación</b>	<b>27,92 días</b>	<b>vie 23/09/11</b>	<b>mié 02/11/11</b>
<b>3 Programación</b>	<b>87,79 días</b>	<b>mar 08/11/11</b>	<b>vie 09/03/12</b>
3.1 Crear código fuente	62,6 días	mar 08/11/11	mié 08/02/12
3.2 Pruebas individuales	21,19 días	mié 08/02/12	vie 09/03/12
<b>3.4 Documentación</b>	<b>2,13 días</b>	<b>mar 08/11/11</b>	<b>vie 11/11/11</b>
<b>4 Integración y Pruebas</b>	<b>16,19 días</b>	<b>vie 09/03/12</b>	<b>lun 02/04/12</b>
4.1 Pruebas de rendimiento	4,69 días	vie 09/03/12	vie 16/03/12
4.2 Pruebas de introducción de datos	5,5 días	vie 16/03/12	lun 26/03/12
4.4 Pruebas del sistema	5,5 días	lun 26/03/12	lun 02/04/12
<b>4.8 Documentación</b>	<b>4,88 días</b>	<b>vie 09/03/12</b>	<b>vie 16/03/12</b>
Cierre proyecto	0 días	lun 02/04/12	lun 02/04/12

Tabla 40. Tareas del Proyecto

## 6.2.2 DIAGRAMA DE GANTT

A continuación se muestra el diagrama de Ganntt junto con las tareas relacionadas y su asignación de recursos:

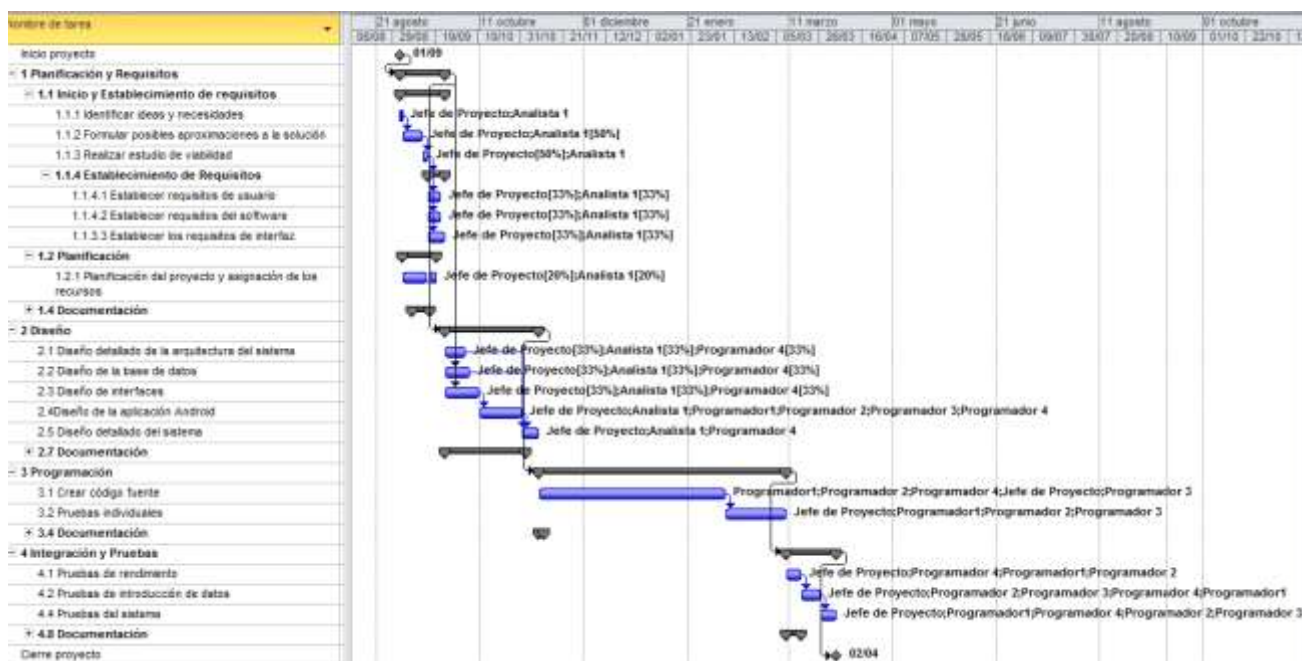


Ilustración 43. Diagrama de Gantt

Como se puede observar algunas tareas se pueden realizar a la vez, por lo tanto se llevan a cabo pero siempre sin sobreasignar a los trabajadores relacionados con ellas.

Este proyecto probablemente se podría haber llevado a cabo con menos personal, pero la duración hubiera sido mayor y en el mercado de aplicaciones para móviles que está en continuo cambio es preferible tardar lo menos posible en sacarla al mercado porque cualquier otra empresa se puede adelantar o la tecnología puede quedar desfasada.

## 6.3 PRESUPUESTO

### 6.3.1 COSTE DEL PERSONAL

Nombre del recurso	Tipo	Capacidad máxima	Tasa estándar	Calendario base	Trabajo	Costo
Jefe de Proyecto	Trabajo	100%	30,00 €/hora	Estándar	1.095,88 horas	35.246,36 €
Analista 1	Trabajo	100%	20,00 €/hora	Estándar	383,08 horas	9.861,61 €
Programador1	Trabajo	100%	15,00 €/hora	Estándar	915,8 horas	13.737,02 €
Programador 2	Trabajo	100%	15,00 €/hora	Estándar	915,8 horas	13.737,02 €
Programador 3	Trabajo	100%	15,00 €/hora	Estándar	876,3 horas	13.144,47 €
Programador 4	Trabajo	100%	15,00 €/hora	Estándar	844,38 horas	12.665,73 €

**Tabla 41. Coste del Personal**

**Total 98.392,21 €**

### 6.3.2 COSTE DEL MATERIAL UTILIZADO

Material	Coste unidad	Cantidad	Total
<b>Ordenador Dell Vostro Desktop 470 MT</b>	649	6	3894
<b>Licencia Microsoft Office 2010 (Hogar y pequeña empresa, 2 equipos)</b>	379	3	1137
<b>Licencia Microsoft Project Standard 2010</b>	775	6	4650
<b>Licencia Microsoft Visio Standard 2010</b>	330	6	1980
<b>Adsl Telefónica (fibra óptica)/mes</b>	39,9	7	279,3
<b>Servidor virtual XL de 1&amp;1 (un año)</b>	279,87	1	279,87
<b>Samsung Galaxy S plus libre (Amazon)</b>	266,08	3	798,24
<b>Samsung Galaxy Mini libre (Amazon)</b>	178,97	3	536,91
<b>Total</b>			<b>13555,32</b>

**Tabla 42. Coste del Material**

En este apartado se han desarrollado los costes de materiales necesarios para desarrollar el proyecto desde cero pero se debería tener en cuenta que algunos de ellos son amortizables.



### 6.3.3 PRESUPUESTO FINAL

El presupuesto final será la suma de los costes del personal y los costes de los materiales utilizados. Por lo tanto:

Costes del personal	Costes del Material	Total
98.392,21 €	13555,32 €	111.947,53 €

Tabla 43. Presupuesto Final

# CAPÍTULO 7. FUTURAS LINEAS DE TRABAJO Y CONCLUSIONES

## 7.1 FUTURAS LINEAS DE TRABAJO

La principal línea de trabajo sería la compra de un dominio y un servidor para la creación de un portal web que de soporte a la aplicación, este portal serviría para la consulta de las quejas y para la creación de un canal RSS que se pudieran compartir en otras páginas.

A parte de la principal línea de trabajo existen otras tres líneas que ayudarían a mejorar el proyecto: un acuerdo con medios de comunicación, colaboración con las administraciones competentes y fomento del uso de la aplicación mediante un sistema de recompensas al usuario.

### 7.1.1 CREACIÓN DEL PORTAL WEB

Como se ha dicho en la introducción de este capítulo ésta sería la principal línea de trabajo. Durante el desarrollo de este proyecto se ha llevado a cabo en remoto con un servidor instalable en el propio ordenador, esto es útil para su etapa de desarrollo, pero no si el proyecto crece o se quiere dar a conocer.

La compra de un dominio y la creación del portal no deberían llevar mucho esfuerzo debido a que la estructura ya esta implementada, se podría reutilizar código y como se ha dicho al principio se quitarían funcionalidades y solo se dejarían las de consulta. Por lo tanto lo que más trabajo llevaría sería portar el código a la web, y diseñar la interfaz.

### 7.1.2 ACUERDOS CON MEDIOS DE COMUNICACIÓN Y FUENTE RSS

Hoy en día para que algo tenga bastante repercusión social es prácticamente obligatorio que aparezca en un medio de comunicación. Como este era uno de los objetivos del proyecto, sería de un gran interés llegar a algún tipo de acuerdo con algún medio de comunicación y que incluyera en su página un lector de RSS de nuestra fuente. Esto sería favorable para las dos

partes debido a que por un lado ellos tendrían una fuente que les proporciona información e incluso la posibilidad de realizar reportajes (como es el caso de Ponle Freno analizado en el estado del arte) y por otro lado nuestra plataforma se daría a conocer.

### **7.1.3 COLABORACIÓN CON LAS ADMINISTRACIONES**

En una situación ideal en la que las autoridades quisieran colaborar con nuestra plataforma, se podría llevar a cabo una serie de reformas en la aplicación las cuales añadirían mayor utilidad a esta creando prácticamente un canal directo entre un ciudadano y su administración.

La primera sería crear una tabla con los correos de las administraciones asociadas a la ciudad que es de su competencia. De esta manera se crearía un nuevo perfil en la aplicación que sería el de la Administración.

A parte de esta modificación se añadiría a la tabla de quejas un nuevo campo el cual podría adquirir los siguientes valores (no comunicado, comunicado, en reparación o reparado). De esta manera un ciudadano una vez realiza la queja tendría la posibilidad de a través de la aplicación mandarle un correo a la administración comunicándole su queja. Una vez hecho esto la administración tomaría las decisiones necesarias y podría mandar reparar la queja del ciudadano y entonces podría utilizar nuestra aplicación para cambiar el estado de la queja a “en reparación” y una vez estuviera reparada, también la administración podría cambiar la queja a “reparada”.

### **7.1.4 SISTEMA DE RECOMPENSAS PARA EL USUARIO**

Para fomentar el uso de la aplicación se podría incorporar un sistema de trofeos para el usuario de tal manera que cuando el usuario consiguiera un determinado logro (por ejemplo que su queja reciba mas de cien apoyos) recibiera un trofeo virtual que pudiera consultar en su perfil.

Para incluir esto se deberían realizar las siguientes modificaciones:

- Crear una tabla de trofeos en la cual una vez un usuario consiguiera un determinado trofeo se insertara en esa tabla el trofeo y el nombre del usuario.
- Añadir una funcionalidad a la aplicación en la cual el usuario pudiera consultar sus datos y sus trofeos.
- Realizar consultas después de realizar una inserción en la base de datos para ver si se ha conseguido un determinado logro o no.

La elección de los trofeos deberá de ser debidamente analizada para que se fomente el buen uso de la aplicación y no un uso indiscriminado que podría comprometer la veracidad de los datos de la aplicación.



Estos sistemas de recompensa ya se han utilizado en distintas plataformas, como en los juegos de videoconsolas, y se ha podido observar cómo ha crecido considerablemente las horas de utilización por parte del usuario.



## CAPÍTULO 8. REFERENCIAS BIBLIOGRAFÍA

Y

- [1] *Blog de programación. Jon Segador.* Disponible [Internet]:  
<<http://www.sgoliver.net/blog/?p=1949>>
- [2] *Manual PHP.* Disponible [Internet]:< <http://www.php.net/manual/es/>>
- [3] *Manual MySQL.* Disponible [Internet]:< <http://dev.mysql.com/doc/refman/5.0/en/>>
- [4] *Entendiendo la interfaz de usuario Android.* Disponible [Internet]:  
<<http://mobiforge.com/designing/story/understanding-user-interface-android-part-3-more-views/>>
- [5] *Como programar en Android.* Disponible [Internet]:  
<<http://blog.findemor.es/2011/02/como-programar-en-android-guia-4/>>
- [6] *Comparativa Android Market vs App Store.* Disponible [Internet]:<<http://www.elibrogordodepetete.com/2012/02/android-market-vs-app-store-comparativa.html>>
- [7] *Función distancia geográfica.* Disponible [Internet]:  
<<http://www.tufuncion.com/distancia-coordenadas>>
- [8] *Manejando longpress.* Disponible [Internet]: <<http://www.kind-kristiansen.no/2011/android-handling-longpresslongclick-on-map-revisited/>>
- [9] *Buscar dirección en mapa de Google.* Disponible [Internet]:  
<<http://jmrogado.wordpress.com/2011/04/24/android-buscar-direccion-mapa/>>
- [10] *Mowes Portable.* Disponible [Internet]:  
<<http://www.chsoftware.net/en/mowes/mowesportable/mowes.htm>>
- [11] *Manual sql. W3schools.* Disponible [Internet]:  
<<http://www.w3schools.com/sql/default.asp>>
- [12] *Manual PHP. W3schools.* Disponible [Internet]:  
<<http://www.w3schools.com/php/default.asp>>



- [13] *API Picasa Web Albums.* Disponible [Internet]: <<https://developers.google.com/picasa-web/>>
- [14] *API Google Maps.* Disponible [Internet]: <<https://developers.google.com/maps/>>
- [15] *API Flickr.* Disponible [Internet]: <<http://www.flickr.com/services/api/>>
- [16] *Fotodenuncia.org.* Disponible [Internet]: <<http://fotodenuncia.org/>>
- [17] *Fotodenuncias.net.* Disponible [Internet]: <<http://www.fotodenuncias.net/>>
- [18] *Antena 3 Ponle Freno.* Disponible [Internet]: <<http://www.antena3.com/ponlefreno/propuestas/>>
- [19] *Apache Server.* Disponible [Internet]: <<http://www.apache.org/>>
- [20] *Conexión Android Picasa Web.* Disponible [Internet]: <<http://www.android-spa.com/viewtopic.php?t=1563>>
- [21] *Video Introductorio Android.* Disponible [Internet]: <<http://www.xtensivearts.com/2009/07/03/episode-1-getting-started/>>
- [22] *Cargar imagen externa en una aplicación Android. Jon Segador.* Disponible [Internet]: <<http://jonsegador.com/2010/03/mostrarcargar-imagen-externa-en-una-aplicacion-android/>>
- [23] *Descarga Java SE.* Disponible [Internet]: <<http://www.oracle.com/technetwork/java/javase/downloads/index.html>>
- [24] *Los 4 fantásticos de la comunicación. Jarboleya.* Disponible [Internet]: <<http://jarboleya.com/2007/08/31/los-4-fantasticos-de-la-comunicacion-blogs-foros-wikis-y-redes-sociales/>>
- [25] *Descarga Android SDK.* Disponible [Internet]: <<http://developer.android.com/sdk/index.html>>
- [26] *Eclipse.* Disponible [Internet]: <<http://www.eclipse.org/>>
- [27] *Tienda Microsoft Office.* Disponible [Internet]: <<http://office.microsoft.com/es-es/home-and-business/>>
- [28] *Dell.* Disponible [Internet]: <<http://www.dell.com>>





- [29] *1and1.es* Disponible [Internet]: <<http://www.1and1.es/>>
- [30] *Especificaciones Samsung Galaxy S. Xakata.* Disponible [Internet]:  
<<http://www.xataka.com/gadgets/telefono-movil/samsung-galaxy-s>>
- [31] *Especificaciones Samsung Galaxy Ace. Xakata.* Disponible [Internet]:  
<<http://www.xataka.com/gadgets/telefono-movil/samsung-galaxy-ace>>
- [32] *Samsung Galaxy S. Samsung* Disponible [Internet]:  
<<http://www.samsung.com/mx/consumer/mobile-phones/mobile-phones/telcel/GT-I9000HKTCE>>
- [33] *Samsung Galaxy Ace. Samsung* Disponible [Internet]:  
<http://www.samsung.com/es/consumer/mobile-phone/smartphones/galaxy/GT-S5830OKAFOP>
- [34] *Android Developers.* Disponible [Internet]:  
<<http://developer.android.com/develop/index.html>>



## CAPÍTULO 9. APÉNDICES

### 9.1 HERRAMIENTAS Y TECNOLOGIAS UTILIZADAS

#### 9.1.1 MYSQL

MySQL es un sistema de gestión de bases de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones. MySQL es un sistema de administración relacional de bases de datos. Una base de datos relacional archiva datos en tablas separadas en vez de colocar todos los datos en un gran archivo. Esto permite velocidad y flexibilidad. Las tablas están conectadas por relaciones definidas que hacen posible combinar datos de diferentes tablas sobre pedido.

MySQL es software de fuente abierta. Fuente abierta significa que es posible para cualquier persona usarlo y modificarlo. Cualquier persona puede bajar el código fuente de MySQL y usarlo sin pagar. Cualquier interesado puede estudiar el código fuente y ajustarlo a sus necesidades. MySQL usa el GPL (GNU General Public License) para definir qué puede hacer y qué no puede hacer con el software en diferentes situaciones.

#### 9.1.2 SERVIDOR APACHE

El servidor HTTP Apache es un servidor web HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras que implementa el protocolo HTTP/1.1 y la noción de su sitio virtual.

El servidor Apache se desarrolla dentro del proyecto HTTP Server (httpd) de la Apache Software Foundation.

Apache presenta entre otras características altamente configurables, bases de datos de autenticación y negociado de contenido, pero tiene el defecto de no tener una interfaz gráfica que ayude a su configuración.

#### 9.1.3 MOWES PORTABLE

Mowes Portable es un software libre que no necesita instalación y que es de fácil manejo. Para descargarlo únicamente hay que navegar por su página seleccionando los elementos que se quieren incluir (en este caso servidor Apache, PHP y MySQL), y el solo crea



un zip que al descomprimirlo se puede utilizar como si fuera un programa que se hubiera instalado previamente.

### 9.1.4 GOOGLE MAPS

Google Maps es el nombre de un servicio de Google. Es un servidor de aplicaciones de mapas en la Web. Ofrece imágenes de mapas desplazables, así como fotos satelitales del mundo e incluso la ruta entre diferentes ubicaciones o imágenes a pie de calle Google Street View.

### 9.1.5 PICASSA WEB SERVICE

Los Álbumes web de Picasa es un sitio web para compartir fotos de Google, a menudo comparado con Flickr y otros sitios similares. La URL de acceso es <http://www.picasaweb.com>.

Permite a los usuarios con una cuenta de Google almacenar y compartir hasta 1 GB de fotos de gran tamaño de forma gratuita. El almacenamiento es ilimitado para fotos de 2048x2048 píxeles o menores para usuarios Google+ y 800x800 para las fotos para todos los demás usuarios. Los vídeos de menos de 15 minutos de duración tampoco cuentan para el límite. Después de que se alcance el límite, las fotos se redimensionan automáticamente.

### 9.1.6 PHP

PHP es un acrónimo recursivo que significa (PHP Hypertext Pre-processor). El gran parecido que posee PHP con los lenguajes más comunes de programación estructurada, como C y Perl, permiten a la mayoría de los programadores crear aplicaciones complejas con una curva de aprendizaje muy corta. También les permite involucrarse con aplicaciones de contenido dinámico sin tener que aprender todo un nuevo grupo de funciones.

PHP es un lenguaje de programación interpretado (Lenguaje de alto rendimiento), diseñado originalmente para la creación de páginas web dinámicas. Se usa principalmente para la interpretación del lado del servidor creando así una manera mas segura de acceder a él.

### 9.1.7 ECLIPSE

Eclipse es un entorno de desarrollo integrado de código abierto multiplataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido", opuesto a las aplicaciones "Cliente-liviano" basadas en navegadores. Esta plataforma, típicamente ha sido usada para desarrollar entornos de desarrollo integrados (del inglés IDE), como el IDE de Java llamado Java Development Toolkit (JDT) y el compilador (ECJ) que se entrega como parte de



Eclipse (y que son usados también para desarrollar el mismo Eclipse). Sin embargo, también se puede usar para otros tipos de aplicaciones cliente, como BitTorrent o Azureus.

Eclipse es también una comunidad de usuarios, extendiendo constantemente las áreas de aplicación cubiertas. Un ejemplo es el recientemente creado Eclipse Modeling Project, cubriendo casi todas las áreas de Model Driven Engineering.

Eclipse fue desarrollado originalmente por IBM como el sucesor de su familia de herramientas para VisualAge. Eclipse es ahora desarrollado por la Fundación Eclipse, una organización independiente sin ánimo de lucro que fomenta una comunidad de código abierto y un conjunto de productos complementarios, capacidades y servicios.

Eclipse fue liberado originalmente bajo la Common Public License, pero después fue re-licenciado bajo la Eclipse Public License. La Free Software Foundation ha dicho que ambas licencias son licencias de software libre, pero son incompatibles con Licencia pública general de GNU (GNU GPL).

### **INSTALACION DE SDK DE ANDROID**

En este apartado se mostrarán los pasos instalación de Eclipse y del sdk de Android para desarrollar aplicaciones

#### **INSTALACION DE ECLIPSE**

Se accede a <http://eclipse.org/downloads/> y se descarga el eclipse IDE for java EE Developers en la versión 32 o 64 bits, la que más convenga al tipo de sistema operativo y ordenador.

Una vez descargada, la se descomprime en el directorio que se desee.

#### **INSTALACION DE JDK VERSION 6**

Para ello se accede a la página <http://www.oracle.com/technetwork/java/javase/downloads/index.html> y se descarga java SE 6 Update 31 (como mínimo). Simplemente se instala y si el sistema pide reiniciar, se reinicia el ordenador.

#### **INSTALACION DE SDK DE ANDROID**

Se accede <http://developer.android.com/sdk/index.html> y se descarga el SDK que se desee para PC, Mac o Linux.

Una vez descargado, se procede a descomprimir su contenido. Se puede hacer en cualquier ubicación, no obstante se debe recomendar que para los usuarios de PC se instale en



**C:/Android/android-sdk\_windows**, creando un directorio Android, y luego descomprimiendo el contenido del zip.

#### INSTALACION DEL COMPLEMENTO ADT PARA ECLIPSE

Se abre ECLIPSE y se va a Help, y luego a Install new Software.

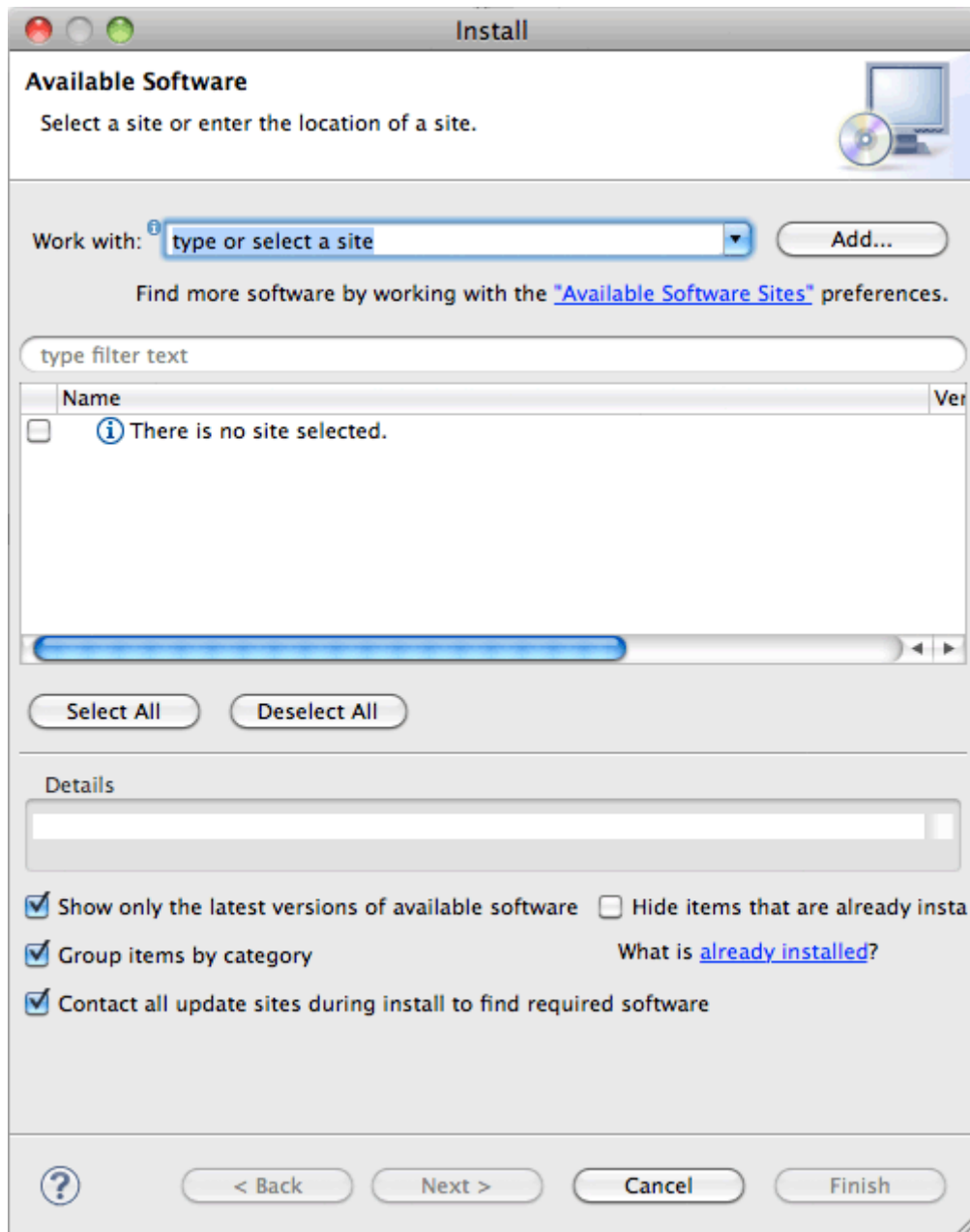


Ilustración 44. ADT Android 1

Se pincha en “add”, y luego se añade la ruta “<https://dl-ssl.google.com/android/eclipse>” en el Name.

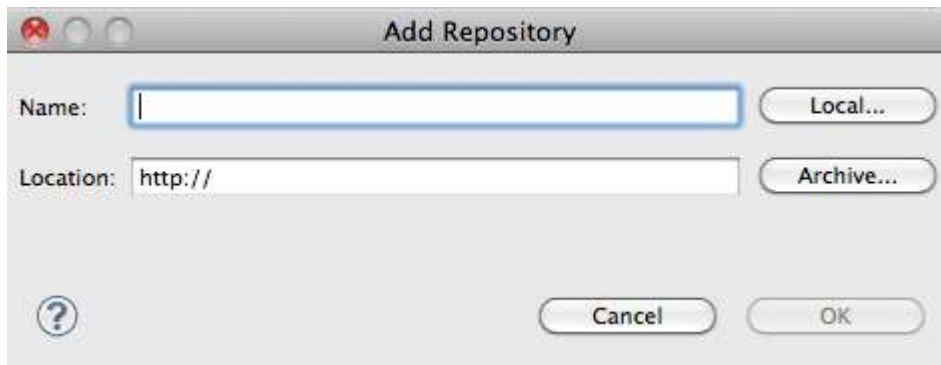


Ilustración 45. ADT Android 2

Cuando realice la búsqueda aparecerá el Developer tools y ya solo queda seleccionarlo y pulsar sobre next.

Aparecerán 4 paquetes y deberá clicar en next. Si alguno del software dice que no tiene firma no pasa nada. Posteriormente se ha de reiniciar ECLIPSE.

Se abre ECLIPSE y se va a preferencias, en la parte izquierda se hace clic en Android y se podrá ver el “SDK location” donde se debe apuntar “C:/Android/android-sdk\_windows” en caso de tener PC, con la opción browse es fácil poner la ubicación.

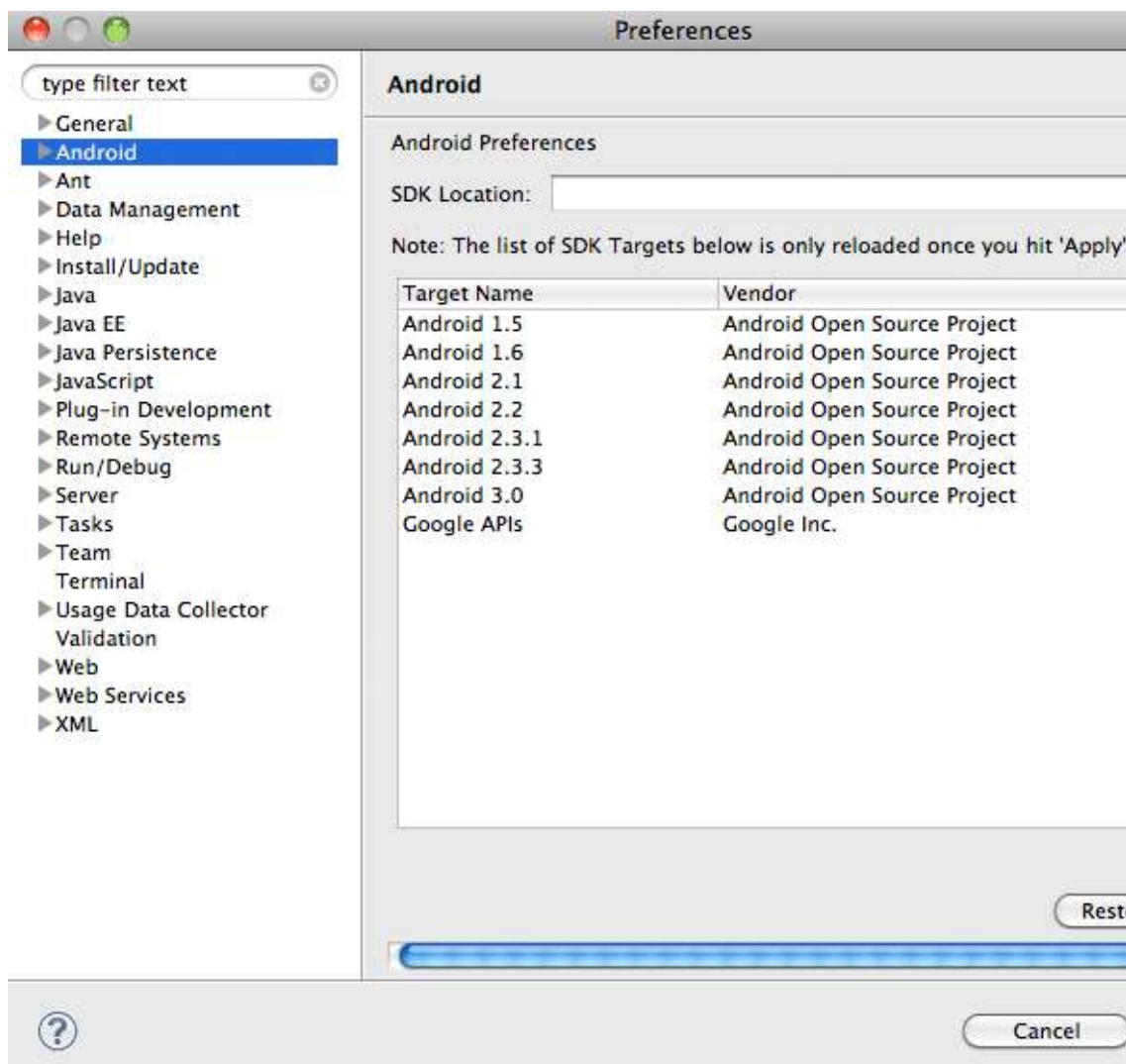


Ilustración 46. ADT Android 3

Se pulsa en “Apply” y después a “OK” y ahora se vuelve a la opción window de ECLIPSE y se pulsa sobre Android SDK Manager.

Se marca todo, son bastantes megas, se pulsa “Install” en los “packages” que aparezcan. Una vez instalado todo se reinicia eclipse y si se va a preferencias de eclipse, a la izquierda de Android, se debería ver las librerías que ya hay disponibles.

Ahora, si se pincha en el ECLIPSE New --> Project , permitirá realizar un proyecto ANDROID.

## OBTENCION DE CLAVE DE GOOGLEMAPS

Para ello, en primer lugar se debe localizar el fichero donde se almacenan los datos del certificado de depuración, llamado “debug.keystore”. Se puede saber la ruta de este fichero accediendo a las preferencias de Eclipse, sección Android, apartado Build (mostrado en la siguiente imagen), y copiar la ruta que aparece en el campo “Default Debug Keystore”:

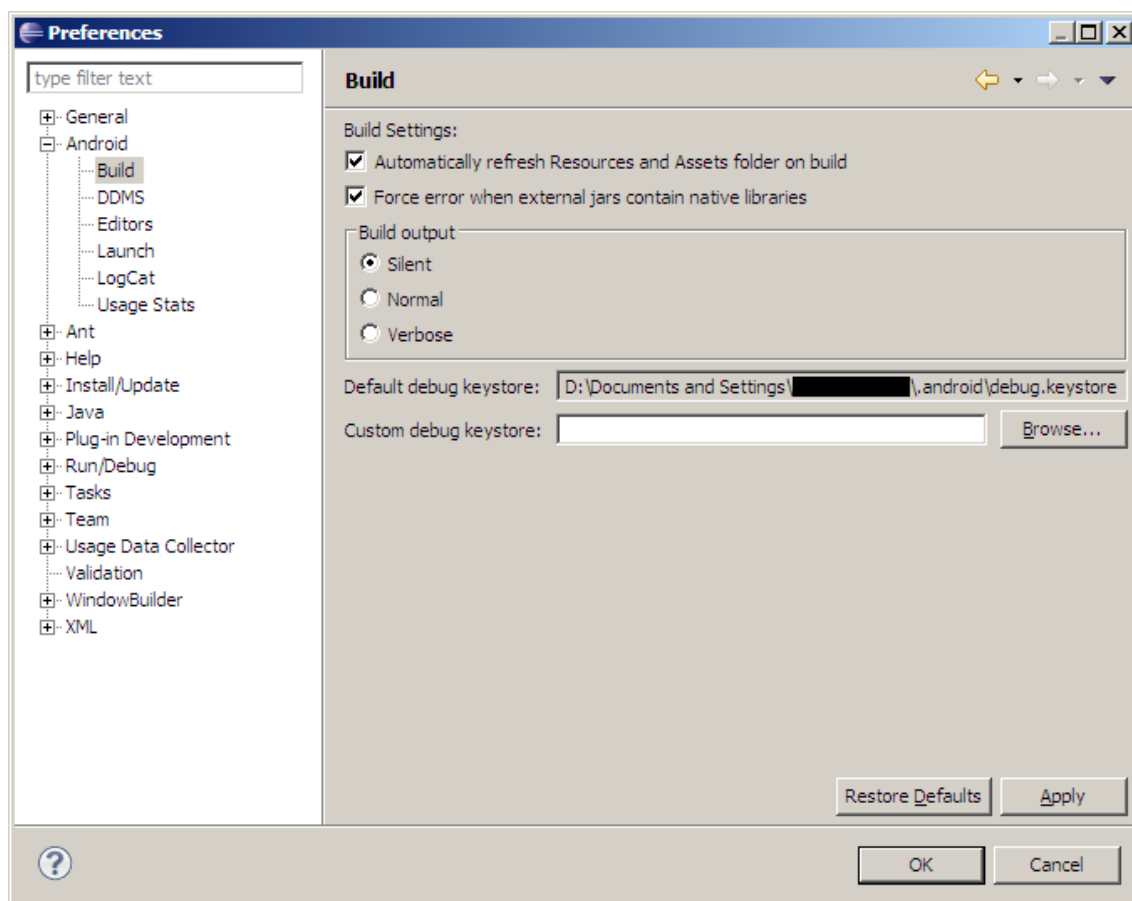


Ilustración 47. ApiKey GoogleMaps 1

Una vez se conoce la localización del fichero “debug.keystore”, se accederá a él mediante la herramienta keytool.exe de java para obtener el *hash* MD5 del certificado. Esto se hará desde la línea de comandos de Windows mediante el siguiente comando:



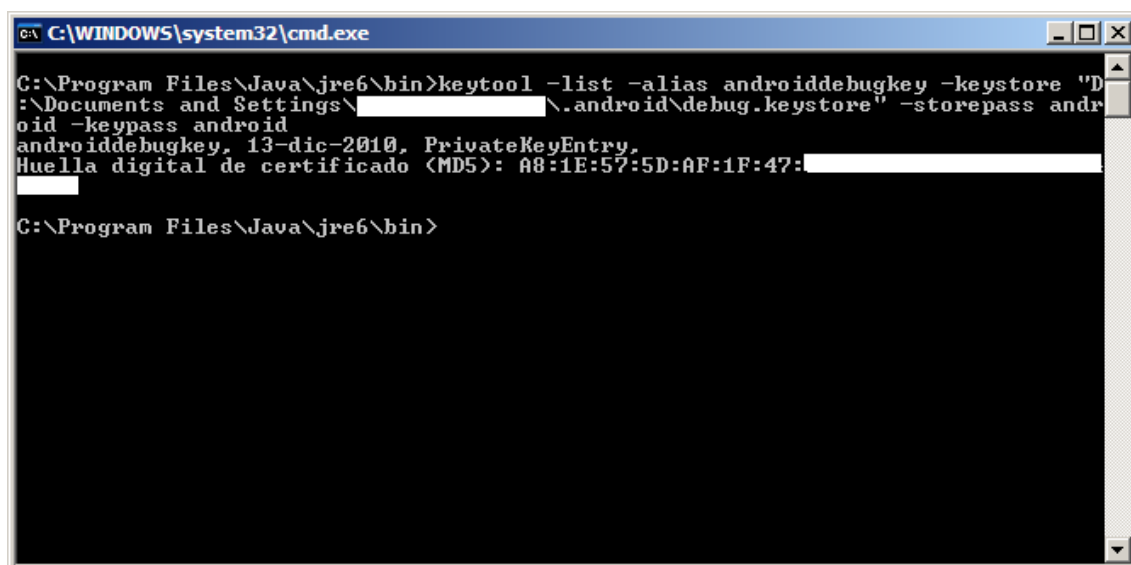


Ilustración 48. ApiKey GoogleMaps 2

Se copia el dato que aparece identificado como “Huella digital de certificado (MD5)” y con éste se accede a la web de Google (<http://code.google.com/android/maps-api-signup.html>) para solicitar una clave de uso de la API de Google Maps para depurar las aplicaciones. Dicha web solicitará la marca MD5 del certificado y proporcionará la clave de uso de la API, como se muestra en la siguiente imagen:

## API de Google Maps

[Página principal de Google Code](#) > [API de Google Maps](#) > [Suscripción al API de Google Maps](#)

### Gracias por suscribirte a la clave del API de Android Maps.

Tu clave es:

0ss-5q6s3FKYk3atMUH

Esta clave es válida para todas las aplicaciones firmadas con el certificado cuya huella dactilar sea:

A8:1E:57:5D:AF:1F:47:

Incluimos un diseño xml de ejemplo para que puedas iniciarte por los senderos de la creación de mapas:

```
<com.google.android.maps.MapView
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:apiKey="0ss-5q6s3FKYk3atMUH"
/>
```

Consulta la [documentación del API](#) para obtener más información.

Ilustración 49. ApiKey GoogleMaps 3



### 9.1.8 SAMSUNG GALAXY S

#### *ESPECIFICACIONES TECNICAS*

- Redes: 2.5G (GSM/ GPRS/ EDGE) : 850 / 900 / 1800 / 1900 MHz; 3G (HSDPA 7.2Mbps, HSUPA 5.76Mbps) : 900 / 1900 / 2100 MHz.
- SO: Android 2.3 (Gingerbread).
- Pantalla: 4.0" WVGA SUPER AMOLED (800×480px) con mDNie.
- Cámara: Cámara 5.0 mega-pixel + VGA Video Telephony Camera, Auto Focus, Auto disparo, "Action shot", "Agrégame", Stop motion, Disparo Cartoon, Disparo sonrisa, Disparo panorámico.
- Video: HD(720p@30fps) en reproducción y grabación de video, Codecs: mpeg4, H.264, H.263, H263Sorenson, DivX HD/ XviD, VC-1; Formatos: 3gp (mp4), WMV (asf), AVI (divx), MKV, FLV.
- Audio: MP3, AAC, AAC+, eAAC+, WMA, OGG, AMR-NB, WAV, MID, AC3, IMY, FLAC, XMF.
- Características de valor añadido: Mensajería integrada 'Social Hub'; Android Market con aplicaciones y contenidos; A-GPS; Realidad Aumentada con Layar Reality Browser; Procesador de aplicaciones de 1 GHz; SMS/ MMS/ Email/ Video Mensajes/ Intercambio con ActiveSync; Sensores: Acelerómetro, Brújula Digital, Sensor de proximidad, Sensor de Luz; Modo "Offline y sin SIM"; Comandos por voz; Lector RSS; Widgets; Seguridad inteligente.
- Conectividad: Tecnología Bluetooth v 3.0, USB v2.0 (High-Speed), Wi-Fi 802.11 b/g/n.
- Memoria: 16GB / 8GB, slot de memoria externa (hasta 32GB).
- Tamaño: 64.2 × 122.4 × 9.9mm, 118g.
- Batería: 1500 mAh

#### *MODIFICACIONES REALIZADAS*

El Samsung Galaxy S utilizado esta flasheado y rooteado y lleva una versión de sistema operativo 4.0.3 (Ice Cream Sandwich).



#### 9.1.9 SAMSUNG GALAXY ACE

##### *ESPECIFICACIONES TECNICAS*

- Procesador: 800 MHz.
- RAM: 278 MB disponible para el usuario.
- Memoria: 158 MB, ampliable mediante tarjetas SD.
- Pantalla: 320X480 px (TFT).
- Dimensiones: 112,4 x 59,9 x 11,5 mm.
- Peso: 113 gramos.
- Otros: Android 2,2 Froyo, cámara de 5 Mpx, batería de 1350 mAh, radio FM con RDS

#### 9.1.10 MICROSOFT OFFICE

Microsoft Office es un paquete de programas informáticos para oficina desarrollado por Microsoft Corp. Se trata de un conjunto de aplicaciones que realizan tareas ofimáticas, es decir, que permiten automatizar y perfeccionar las actividades habituales de una oficina.

#### 9.1.11 MICROSOFT PROJECT

Microsoft Project (o MSP) es un software de administración de proyectos diseñado, desarrollado y comercializado por Microsoft para asistir a administradores de proyectos en el desarrollo de planes, asignación de recursos a tareas, dar seguimiento al progreso, administrar presupuesto y analizar cargas de trabajo.

#### 9.1.12 MICROSOFT VISIO

Microsoft Visio es un software de dibujo vectorial para Microsoft Windows. Visio comenzó a formar parte de los productos de Microsoft cuando fue adquirida la compañía Visio en el año 2000.

Las herramientas que lo componen permiten realizar diagramas de oficinas, diagramas de bases de datos, diagramas de flujo de programas, UML, y más, que permiten iniciar al usuario en los lenguajes de programación.

